

THESIS / THÈSE

MASTER EN SCIENCES INFORMATIQUES

Logiciel graphique de manipulation interactive de surfaces 3-D

Hucq, B.

Award date:
1984

Awarding institution:
Université de Namur

[Link to publication](#)

General rights

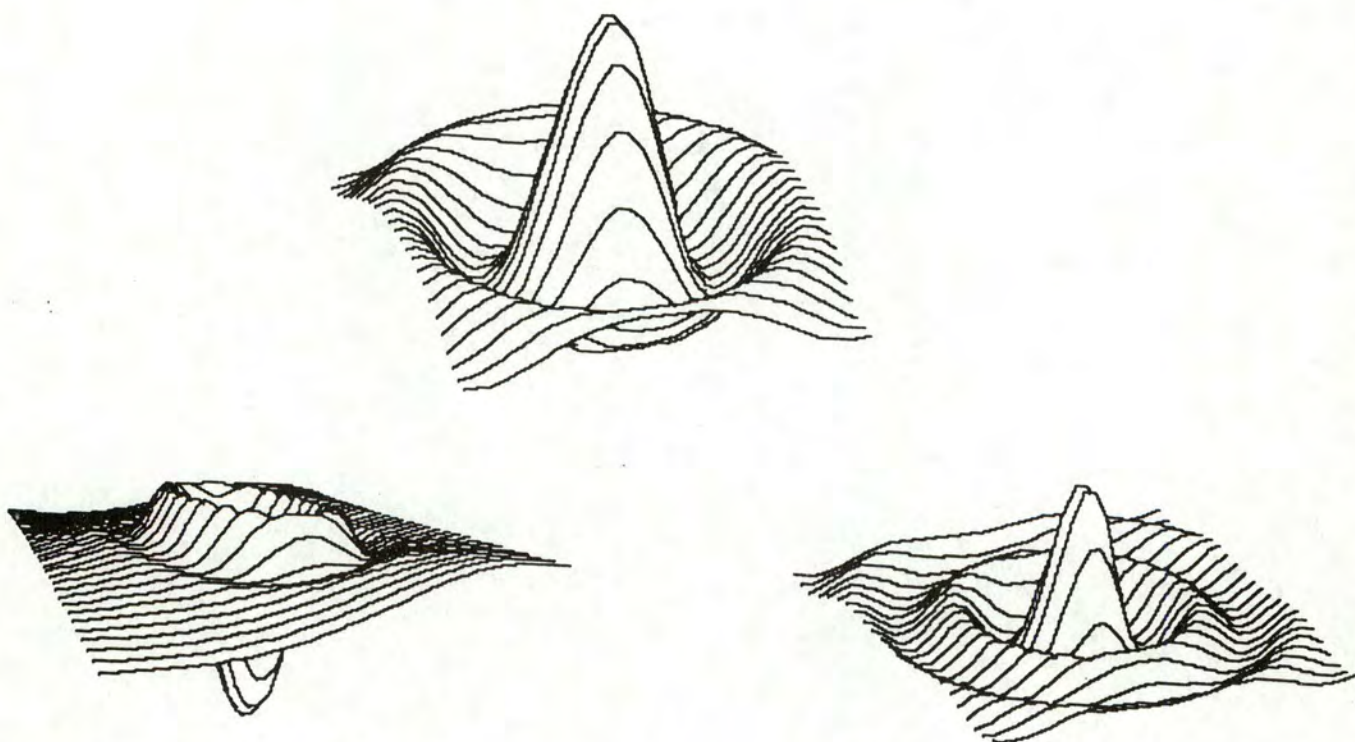
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

LOGICIEL GRAPHIQUE DE MANIPULATION INTERACTIVE DE SURFACES 3-D.



B. HUCQ 1984

MEMOIRE PRÉSENTÉ EN VUE DE
L'OBTENTION DU GRADE DE
LICENCIÉ ET MAÎTRE EN
INFORMATIQUE.

Si par essence, un mémoire est un travail individuel, je ne peux oublier l'aide indispensable de Messieurs Ramaekers et Michel, directeurs, éclaireurs et conseillers, ainsi que celle de Messieurs Beguin, Donnay et Vercauteren.

Enfin, pour leur soutien moral et dactylographique, je me dois de remercier encore mes Parents et Françoise.

TABLE DES MATIERES.

<u>INTRODUCTION</u>	1.
<u>PREMIERE PARTIE : Méthode de conception d'un logiciel graphique interactif.</u>	3.
<u>CHAPITRE I : Le logiciel graphique interactif : une démarche de conception.</u>	3.
I.1 Les particularités du logiciel graphique interactif.	5.
I.1.1 Des objectifs au fonctionnement d'un LGI.	
I.1.1.1 L'image graphique.	
I.1.1.2 Le fonctionnement général d'un LGI.	
I.1.1.2.1 L'"output process".	
I.1.1.2.2 L'"input process".	
I.1.2 L'interface Homme-Machine d'un LGI.	
I.1.3 Les conséquences de la diversité des moyens techniques d'entrée/sortie.	
I.1.4 Conclusion.	
I.2 Méthodologie de Conception :Premières approches.	17.
I.2.1 Vers une démarche classique de développement de logiciels.	
I.2.2 Les démarches de conception liées aux LGI.	
I.2.2.1 La mise en oeuvre d'un logiciel graphique de base.	
I.2.2.1.1 Le logiciel de mise en forme.	
I.2.2.1.2 Le logiciel graphique de base.	
I.2.2.1.3 Conclusion.	
I.2.2.2 La conception de l'interface Homme-Machine.	
I.3 Une démarche de conception d'un LGI : notre choix.	27.
I.3.1 Les critères de choix.	
I.3.2 Le choix.	
I.4 Conclusion	29.
<u>DEUXIEME PARTIE : Un logiciel graphique interactif: Manipulation interactive de surfaces 3-D.</u>	31.
<u>CHAPITRE II:Du projet cadre aux spécifications globales du logiciel.</u>	31.
II.1 De l'expression d'un besoin aux objectifs du logiciel.	31.
II.1.1 Les besoins	
II.1.2 Les objectifs du logiciel: le projet cadre de notre étude	

II.1.3	La justification de ces objectifs.	
II.2	Cadre général du projet informatique.	33.
II.2.1	Par rapport aux objectifs fonctionnels.	
II.2.1.1	Les différents types de surfaces 3-D et notre choix.	
II.2.1.2	Les grandes fonctions du logiciel.	
II.2.1.2.1	Introduction de la surface.	
II.2.1.2.2	Représentation de la surface.	
II.2.1.2.3	Visualisation de la surface.	
II.2.1.2.4	Manipulation de la surface.	
II.2.2	Par rapport aux objectifs "moyens".	
II.2.3	Par rapport aux objectifs "qualités".	
II.2.4	Par rapport aux objectifs "utilisateurs".	
II.3	Cadre général de l'application géographique.	45.
II.3.1	Les besoins liés à l'application géographique.	
II.3.2	Le projet-cadre de l'application géographique.	
II.3.3	Spécifications globales complémentaires liées à cette application géographique.	
II.3.3.1	Par rapport aux fonctions particulières à l'application géographique.	
II.3.3.1.1	Introduction de la surface.	
II.3.3.1.2	Visualisation de la surface.	
II.3.3.1.3	Modélisation d'une variation réelle.	
II.3.3.1.4	Visualisation d'une variation modélisée.	
II.3.3.2	Remarques de spécification.	
II.4	Cadre général de l'application chimique.	52.
II.4.1	Les besoins liés à l'application chimique.	
II.4.2	Le projet-cadre de cette application chimique.	
II.4.3	Spécifications globales complémentaires liées à l'application chimique.	
II.5	En conclusion.	54.
CHAPITRE III	<u>L'analyse conceptuelle du logiciel général et de ses applications.</u>	55.
III.1	But et méthodologie de l'analyse conceptuelle.	55.
III.2	L'élaboration d'une "solution".	56.
III.2.1	La décomposition en phases.	
III.2.2	La dynamique des phases.	
III.2.3	Objectifs et contraintes des phases.	

III.3 L'analyse conceptuelle.	62.
III.3.1 Elaboration des sous-schémas conceptuels.	
III.3.2 Consolidation des sous-schémas conceptuels.	
III.3.2.1 Au niveau des données.	
III.3.2.2 Au niveau des traitements.	
III.4 Conclusion.	62.
<u>CHAPITRE IV : Une architecture logique globale du logiciel.</u>	<u>64.</u>
IV.1 Introduction.	64.
IV.2 La hiérarchisation du logiciel.	65.
IV.3 La modularisation du logiciel.	66.
IV.3.1 Au niveau 6.	
IV.3.2 Au niveau 5.	
IV.3.3 Au niveau 4.	
IV.3.3.1 Les modules d'accès à la BD.	
IV.3.3.2 Les modules de gestion des moyens techniques d'entrée/sortie.	
IV.3.4 Aux niveaux 3 et 2.	
IV.4 Les spécifications abstraites des modules.	73.
IV.4.1 Rappels méthodologiques.	
IV.4.2 Les spécifications abstraites.	
IV.5 Un graphe d'architecture logique de notre système.	74.
IV.6 Les choix préalables à la poursuite de notre travail.	75.
IV.7 Conclusion.	76.
<u>CHAPITRE V : Spécification de l'interface Homme-Machine.</u>	<u>77.</u>
V.1 But et rappels méthodologiques de la spécification d'un interface.	77.
V.2 Principes d'interfaçage.	79.
V.2.1 Les principes.	
V.2.2 La facilité d'utilisation : les règles générales de sa mise en oeuvre.	
V.2.2.1 Au niveau de l'"user's model".	
V.2.2.2 Au niveau du "command language".	
V.2.2.3 Au niveau de l'"information display".	

V.2.2.4	Au niveau du "feedback".	
V.2.2.5	Généralités sur l'utilisation de l'écran graphique et annexe.	
V.3	Spécification de l'interface.	87.
V.4	Conclusion.	87.
<u>CHAPITRE VI : Conception et mise en oeuvre physique des modules.</u>		88.
VI.1	Rappels méthodologiques.	88.
VI.2	Algorithmes de conception.	
VI.3	Elaboration d'une architecture physique : les choix préalables d'implémentation physique.	88.
VI.3.1	Le système Mégatek.	
VI.3.2	Le LGDB Wand.	
VI.3.3	Le langage de programmation.	
<u>CONCLUSIONS GENERALES</u>		94.
<u>BIBLIOGRAPHIE</u>		99.
<u>ANNEXES</u>		102.
ANNEXE 0	Figures et tableaux.	102.
1)	Liste des figures.	
2)	Liste des tableaux.	
3)	Tableaux et figures hors textes.	
ANNEXE I	Objectifs et contraintes des phases de l'application.	108.
ANNEXE II	Les sous-schémas conceptuels des données et traitements.	115.
ANNEXE III	Spécifications abstraites des modules de l'architecture logique.	175.
ANNEXE IV	Spécification de l'interface.	192.
ANNEXE V	Conception algorithmique.	204.

INTRODUCTION

Le travail réalisé dans le cadre de ce mémoire poursuit deux objectifs majeurs.

Le premier objectif consiste en la réalisation d'une approche méthodologique de conception d'un logiciel graphique interactif. Quel substrat méthodologique classique de conception peut-on choisir pour un tel logiciel?

Quels sont les apports et modifications éventuels à ce substrat pour tenir compte des particularités d'un logiciel graphique interactif?

Le résultat du développement d'une méthodologie de conception se doit d'être conforme à des critères précis et définis au préalable. Ces critères sont les guides de notre recherche et de son évaluation.

Dans la littérature actuelle, on remarque la pauvreté des tentatives d'investigations dans le domaine méthodologique de conception d'un logiciel graphique interactif. Et ceci, tant en qualité (les méthodologies proposées sont partielles) qu'en quantité!

Nous croyons donc toucher là un sujet prometteur.

Le second objectif correspond à l'application de la démarche de conception issue de notre approche méthodologique.

L'application de la démarche se fait dans le cadre d'un logiciel graphique interactif de manipulation de surfaces tri-dimensionnelles (3-D).

Ce type de logiciel selon nos informations n'existe pas en Belgique, notamment par son caractère multi-cibles.

Celui-ci permet la diversité des surfaces 3-D étudiables et manipulables par notre produit logiciel.

Le résultat de l'application de notre démarche est un logiciel conforme à nos attentes mais aussi une évaluation de la démarche entreprise pour concevoir ce logiciel.

Au cours de cette évaluation, on doit répondre aux questions suivantes.

La démarche est-elle vraiment opérationnelle dans le cadre de l'application choisie?

La démarche présente-t-elle des intérêts?

Notre mémoire présente deux parties liées à ses objectifs.

La première partie décrit notre approche méthodologique et ses aboutissements en terme de démarche de conception d'un logiciel graphique interactif.

La seconde partie, plus importante, reprend les résultats de l'application des étapes de conception à la production d'un logiciel de manipulation de surfaces 3-D.

Ière PARTIE

METHODE DE CONCEPTION D'UN
LOGICIEL GRAPHIQUE INTERACTIF

CHAPITRE I

LE LOGICIEL GRAPHIQUE INTERACTIF : UNE DEMARCHE DE CONCEPTION

Nous allons tenter de définir dans ce chapitre une démarche, une méthodologie de conception d'un logiciel graphique interactif (LGI).

Qu'entend-t-on par démarche de conception?

On doit envisager par là une suite d'étapes reliées entre elles et où à chacune d'entre elles, sont associés un ensemble de principes et de règles utilisés pour mettre en oeuvre l'étape. La relation entre les différents stades de la démarche est une relation de précédence où la terminaison des étapes $i-1$ est nécessaire au démarrage des étapes i . On pourra formaliser cette démarche par un diagramme de conception tel un flow-chart d'un algorithme mais où les instructions algorithmiques sont des étapes de conception. On admet donc la possibilité du parallélisme entre plusieurs étapes de conception, les retours en arrière, ... Plus loin, on utilisera le formalisme du modèle de la dynamique (Bodart, 1983) pour exprimer un diagramme de conception.

A propos des principes et règles propres à chaque étape, nous devons ici insister sur la cohérence de conception interne à chaque étape. Ceci signifie que l'ensemble de principes et règles ne peuvent varier au long d'éventuelles exécutions répétées de l'étape de conception. Ainsi, à l'étape de codage, on codera un algorithme par un langage informatique donné; une structure conditionnelle de l'algorithme sera codée en une ligne de code adéquate. Il est donc évident que ce codage ne pourra varier dans sa sémantique: la règle de codage d'une structure conditionnelle ne changera pas.

Qu'attend-t-on de cette démarche de conception?

Tout d'abord, qu'elle soit entr'autres, garante des qualités attendues de tout logiciel. Notons ici que la démarche de conception n'est pas la seule garante des qualités requises. L'ensemble des étapes peut être suivi de façon rigoureuse, chaque étape peut être mise en oeuvre en accord avec les règles et principes prescrits, mais la logique de raisonnement peut

elle être erronée ou partielle et donc conduire à un logiciel, par exemple, peu robuste.

Rappelons les principales qualités d'un logiciel (D'après Van Lamsweerde, cours de Méthodologie de Développement de logiciels, 1983) :

- . Fiabilité : Un logiciel fiable sera à la fois valide (par rapport à ce qu'on attend de lui), robuste (par rapport aux erreurs possibles d'opération) et testable.

- . Maintenance : Un logiciel aura une maintenance aisée si les modifications, les ajouts et les suppressions de fonctions sont possibles, mais aussi relativement faciles.

- . Portabilité : Un logiciel sera portable s'il peut être exécuté sur plusieurs configurations matérielles.

- . Efficacité : Un logiciel sera efficace si les performances du logiciel en exécution sont compatibles aux attentes, qui sont parfois des nécessités.

- . Userfriendly : Un logiciel est "userfriendly" si son utilisation est agréable et son apprentissage aisé.

- . Sécurité : Un logiciel est sûr si des fonctions dont l'usage est réglementé, sont suffisamment protégées envers une fraude quelconque.

On espère ensuite d'une démarche de conception qu'elle soit en accord avec le type de logiciel à développer. Un logiciel de calcul de la factorielle sera élaboré selon une méthode simplifiée (des étapes ne seront pas mises en oeuvre, par exemple les spécifications abstraites ne seront pas nécessaires). Par contre, un logiciel de gestion d'un service commercial d'une entreprise risque d'imposer l'utilisation d'une démarche élaborée et complète. C'est pourquoi, nous présenterons au paragraphe I.1 les particularités d'un LGI afin de mettre en évidence les éléments susceptibles de modifier une méthode classique que nous décrirons au paragraphe I.2.1.

Enfin, une démarche de conception se doit d'être opérationnelle. Il nous semble essentiel et évident qu'une démarche de conception puisse réellement conduire à la conception d'un logiciel!

Ceci implique qu'après avoir défini notre choix de démarche de conception d'un LGI, nous le mettons en oeuvre dans la conception d'un logiciel de manipulation interactive de surfaces 3-D (Deuxième partie du mémoire).

I . 1 Les particularités d'un logiciel graphique interactif.

De l'expression "logiciel graphique interactif", il est aisé de tirer deux particularités évidentes de ce type de logiciel.

Les LGI utilisent un ensemble de périphériques d'entrée-sortie, notamment un support graphique de visualisation, qui permettent une interaction importante et facile entre l'utilisateur et le logiciel.

Nous allons analyser ces deux particularités et leurs conséquences sur une démarche de conception d'un LGI. Mais auparavant, nous allons brosser un tableau général des objectifs et du fonctionnement d'un LGI. Ce tableau ne pourra qu'être général vu la diversité, la complexité et le caractère évolutif du "graphique".

I.1.1 Des objectifs au fonctionnement d'un LGI.

L'objectif de base d'un LGI est de visualiser sur un support (écran graphique - graphic display) une image modifiable par l'interaction humaine. Ainsi, un logiciel de CAO en électronique visualisera l'état actuel d'un circuit électronique, de ses composants et de ses caractéristiques. L'utilisateur par son interaction modifiera cet état. Le LGI devra donc assurer la visualisation de chaque état et permettra l'interaction pour modifier, suivant certaines contraintes, l'état présent.

I.1.1.1. L'image graphique.

On observe tout d'abord différentes catégories d'images engendrées par ordinateur (Lucas, 1982, p9) :

- . images non figuratives utilisées pour des recherches plastiques,
- . images figuratives pour notamment la CAO,
- . graphiques pour représenter rapidement des phénomènes, des relations entre des informations diverses,
- . images animées.

Si l'on envisage une image en détail, on peut remarquer que chaque image est issue de l'association d'éléments liés à l'information à visualiser. De façon plus formelle, Bertin (1979) présente la notion de langage graphique composé d'éléments. Ce langage est fonction de l'application.

Ces éléments d'un langage graphique sont appelés taches et sont caractérisés par leur

- . forme,
- . emplacement sur le support graphique,
- . taille ou surface employée sur le support graphique,
- . orientation par rapport à un système d'axes de référence,
- . grain ou nombre d'éléments séparables dans la tache,
- . intensité ou valeur de gris de la tache,
- . couleur.

L'image est visualisée sur un support graphique. Il existe deux types de supports suivant le type de quadrillage des surfaces de visualisation associées à ces supports. La quadrillage définit un système de coordonnées permettant de repérer tout point accessible par un couple (x,y) . On aura donc (fig.1) :

- . les surfaces telles que le couple (x,y) désigne un sommet du quadrillage
- . les surfaces telles que le couple (x,y) désigne un carreau élémentaire.

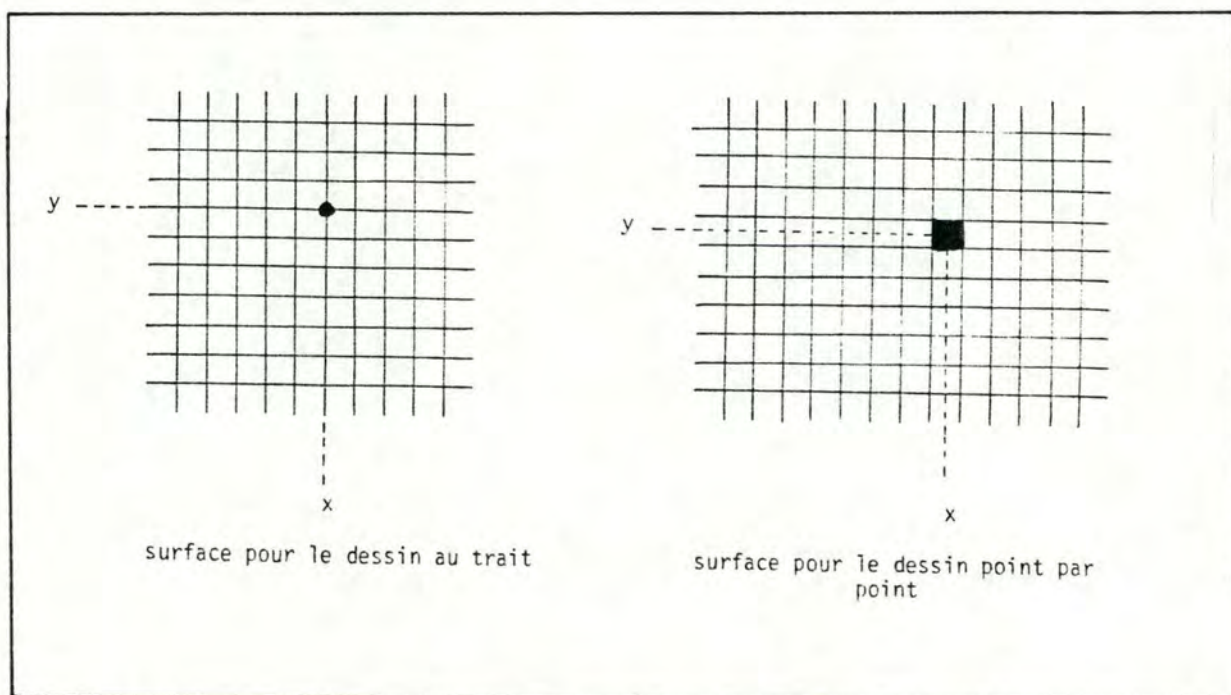


FIGURE 1 : Deux types de surfaces de dessin

(D'après Lucas, 1982, p11).

"Le premier type de surface est conçu essentiellement pour la représentation de dessins au trait, dessins obtenus en joignant un certain nombre de positions par des traits au graphisme varié. Le deuxième type de surface est conçu pour obtenir des images point par point, en définissant la couleur ou la luminosité de chacun des carreaux (pixel)" (Lucas, 1982, p11).

L'image produite sur un écran graphique est fugitive par nature, ce qui permet de la modifier mais implique qu'un dispositif permette d'assurer le balayage constant de l'image, de manière à la laisser affichée le temps voulu. Ce dispositif est appelé mémoire d'entretien.

La régénération d'images doit se faire au moins 25 fois par seconde afin d'éviter un tremblement d'image. Cette fréquence de régénération sera limitée par la quantité d'information à régénérer. L'évolution technologique fut importante dans ce sens: on est passé des centaines de points élémentaires (pixel ou croisement de quadrillage) à plusieurs milliers de lignes de points! (Newman et Sproull, 1979, p10).

Il existe également de façon globale deux types de mémoires d'entretien : . les mémoires d'entretien vives qui contiennent la liste d'affichage. Cette liste d'affichage contient des instructions exécutées par un processeur graphique en permanence, afin de régénérer l'image: "La liste d'affichage peut même dans certains cas refléter la structure du dessin par l'utilisation de sous-dessins(Lucas, 1982, p13). On aura alors une mémoire d'entretien structurée souvent associée aux écrans dits "Line- drawing".

. les mémoires d'entretien à accès direct où l'information contenue reprend la définition point par point de l'image. Ces mémoires seront souvent associées aux écrans dits "Storage- tube".

I.1.1.2. Le fonctionnement général d'un LGI.

Comment assurer la visualisation et l'interaction, en bref, comment atteindre les objectifs du LGI?
La réponse sera évidemment différente d'application en application.

Mais l'on peut tenter de modéliser le processus de visualisation (output process) et d'entrée d'information (input process). Ces deux processus sont insérés dans celui, plus général, d'interaction (interactive process) de la figure 2.

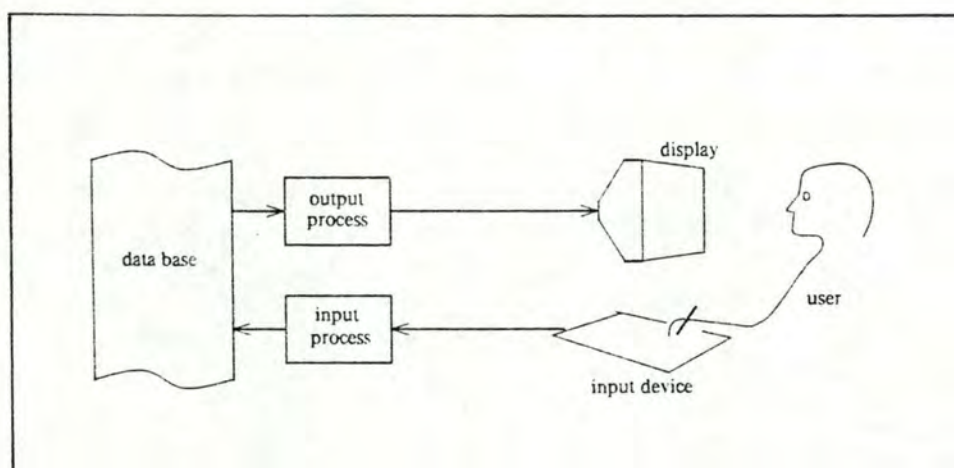


FIGURE 2 : Un modèle simple du processus d'interaction.

(D'après Newman et Sproull, 1979, p464)

On peut décrire ce processus d'interaction par la suite répétée de ces deux événements :

- . L'utilisateur fait parvenir par le biais de l'input process des commandes modifiant par le LGI la BD de l'application.

- . La modification de cette BD modifie par l'output process la liste d'affichage de l'écran graphique et donc modifie l'image.

Ce processus d'interaction est évidemment simplifié.

La figure 3 montre une version proche du modèle de la figure 2 où l'on appréhende le rôle du programme d'application.

Nous verrons plus tard que ce modèle d'interaction pourra être amélioré par la présence de feedbacks à différents niveaux de l'interaction.

Pour approcher de façon plus précise le fonctionnement d'un LGI, nous allons étudier "l'input process" et "l'output process".

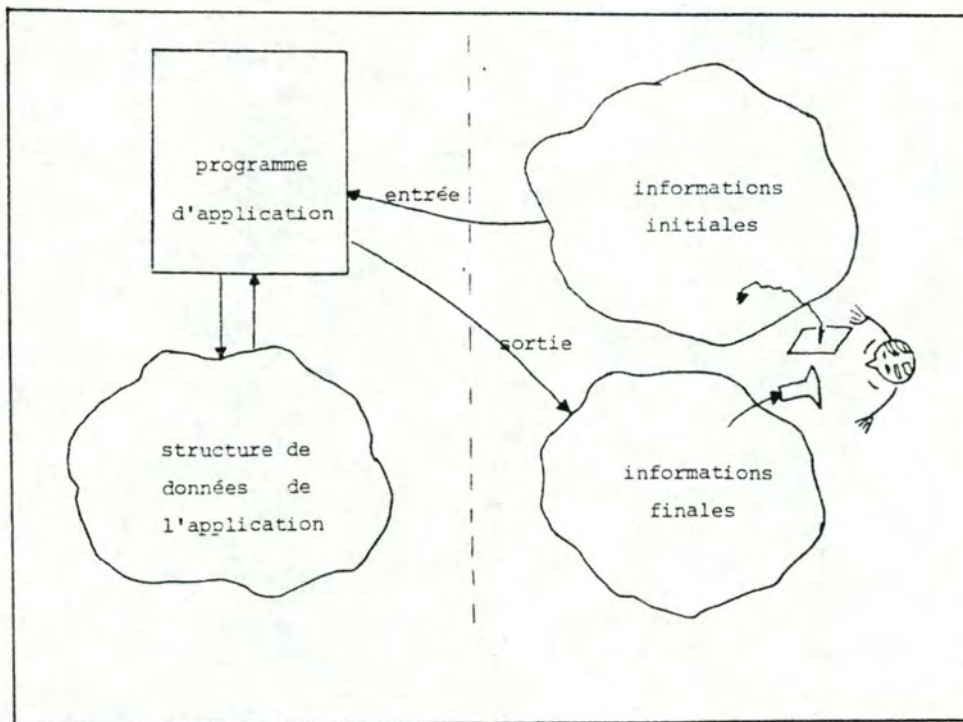


FIGURE 3 : Un modèle simple du processus d'interaction.

(D'après Lucas, 1982, p16).

I.1.1.2.1. L'"output process".

Supposons tout d'abord que le support de visualisation est du type Line-drawing (oscilloscope), où il y a donc une mémoire d'entretien structurée contenant une liste d'affichage.

Le modèle simple du processus de visualisation est donné par la figure 4.

On y remarque que la BD de l'application est à l'origine, par le biais d'un "viewing algorithm", d'une BD restreinte et destinée à la visualisation: c'est la liste d'affichage structurée qui générera l'image.

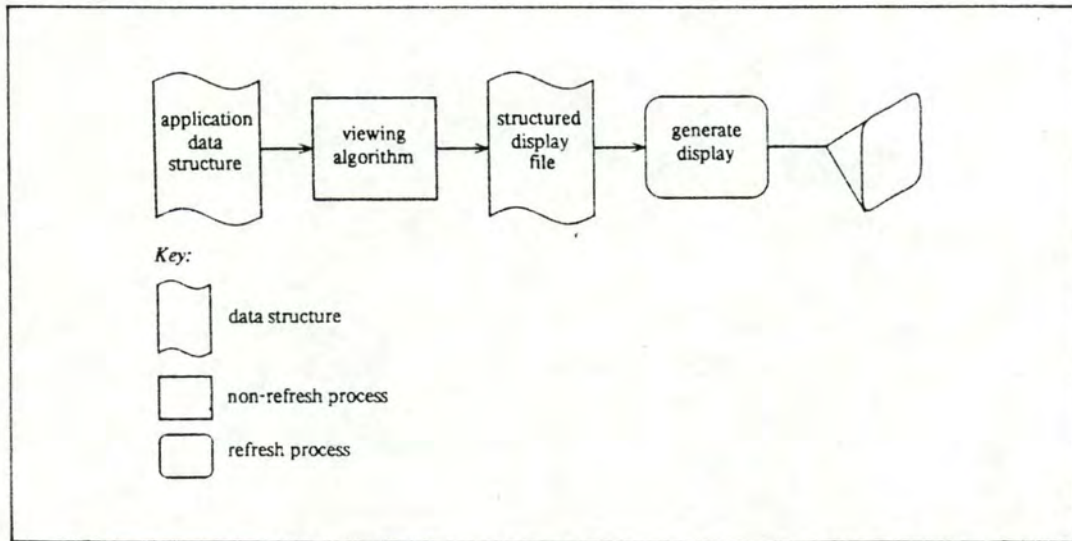


FIGURE 4 : Un modèle simple du processus de visualisation sur surface à trait.

(D'après Newman et Sproull, 1979, p428).

On peut essayer de préciser comment se fait cette génération d'image à partir des instructions de la liste.

La figure 5 montre ainsi le fonctionnement du "generate display".

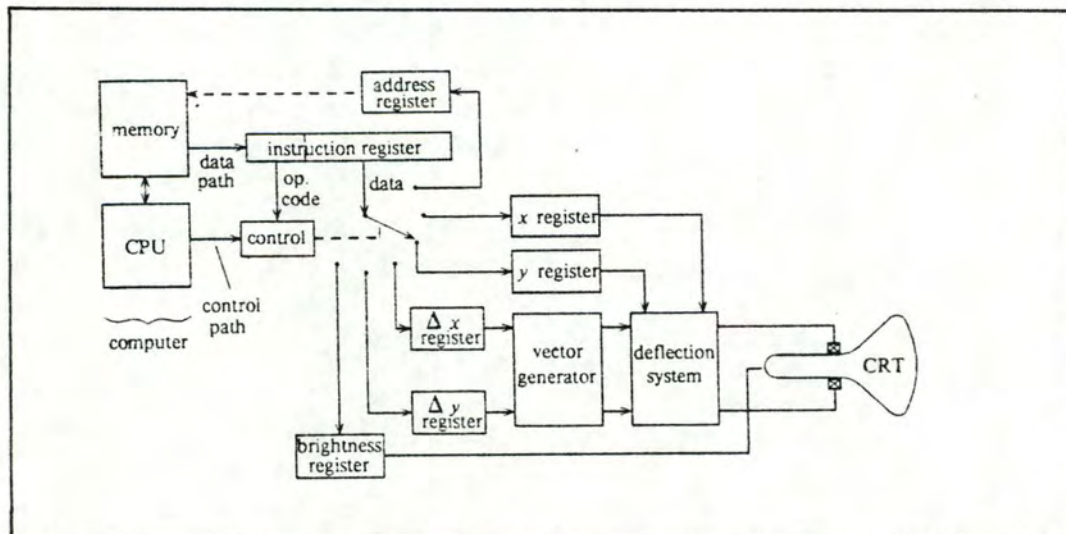


FIGURE 5 : Schéma général de fonctionnement d'un écran "Line-drawing"

(D'après Newman et Sproull, 1979, p415)

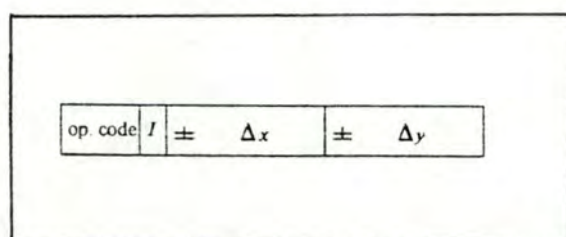


FIGURE 6 : Instruction graphique simplifiée

(D'après Newman et Sproull, 1979, p415).

Les instructions sont du type de la figure 6.

Elles seront exécutées par le module CONTROL qui en décodera le code opératoire. Les opérations possibles seront, dans une version simplifiée,

- . positionnement du faisceau électronique du CRT (Cathodic Ray Tube) par des coordonnées absolues ou relatives.
- . changement de valeur du registre de l'adresse de la prochaine instruction à exécuter.
- . changement de l'intensité de la génération des lignes visualisées.

On pourra ajouter des instructions allant lire par exemple dans une ROM des instructions de génération de caractères préformatés.

Si maintenant, nous supposons que le support de visualisation adopté est du type "Storage tube" où il y a définition complète de l'image pixel par pixel, le modèle de l'"output process" sera celui de la figure 7.

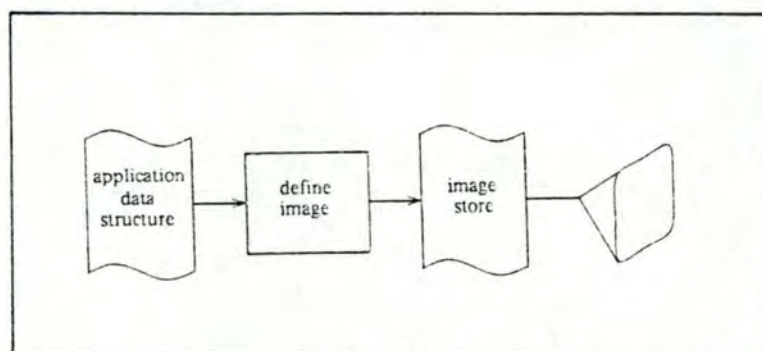


FIGURE 7 : Un modèle simple de processus de visualisation sur surface point à point.

(D'après Newman et Sproull, 1979, p429)

Enfin, l'évolution récente des supports de visualisation montre l'utilisation de modules hardware pour implémenter certaines fonctions du "viewing algorithm" (rotation, translation d'éléments visualisés), ceci dans le but d'une accélération de l'output process.

On remarque donc au delà de la description du processus de visualisation (output process) que celui-ci dépend du moyen technique de visualisation. Nous reparlerons de cette dépendance du LGI vis-à-vis des moyens techniques d'entrée.

I.1.1.2.2 L'"input process".

La fonction d'entrée se caractérise par sa diversité issue d'une part, de la variabilité des actions d'interactions désirées, d'autre part, des moyens techniques mis en oeuvre pour réaliser ces actions.

On peut modéliser le processus d'entrée par la figure 8.

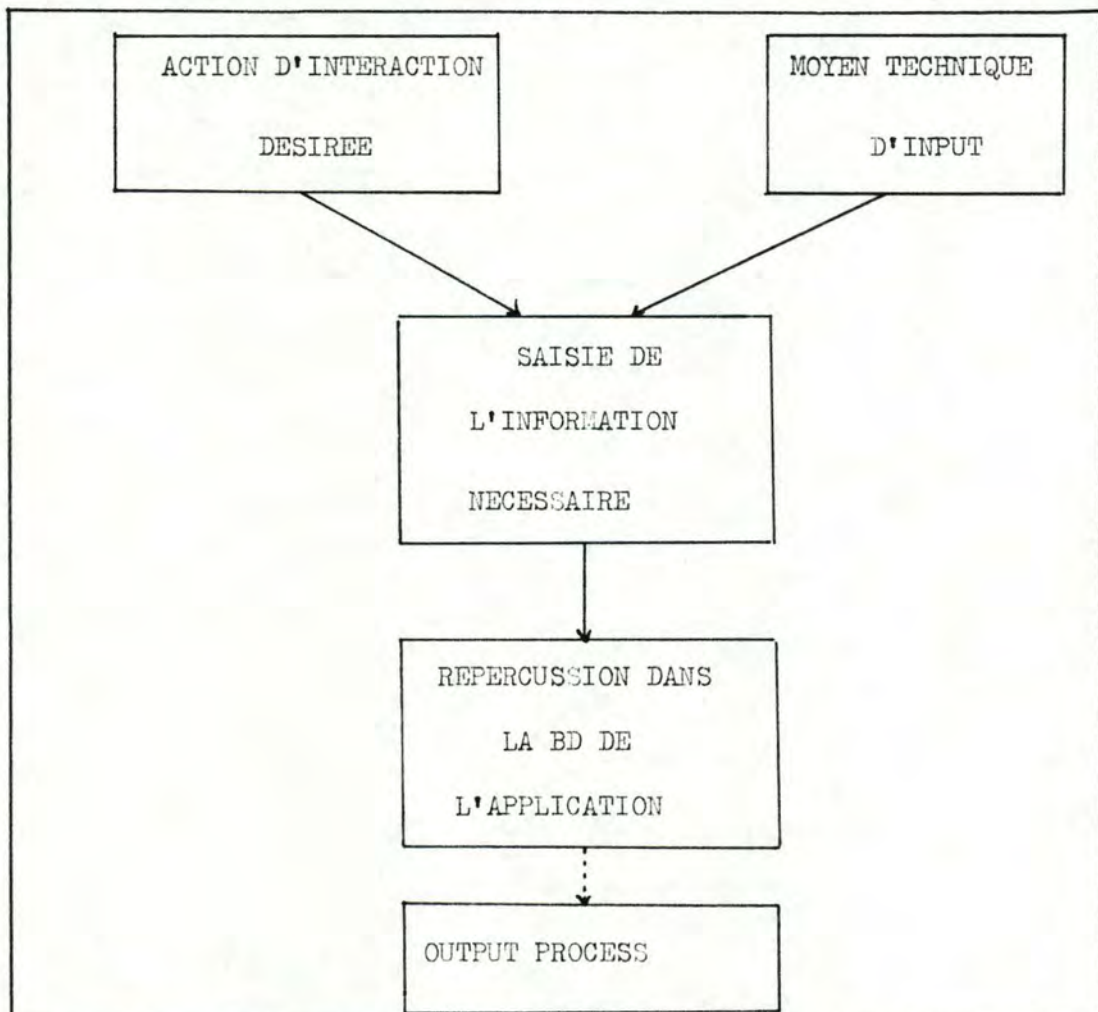


FIGURE 8 : Un modèle simple de processus d'entrée.

On peut identifier ces différentes étapes en prenant l'exemple suivant:
 "On veut sélectionner avec la "souris" un item du menu actuel et ce,
 pour visualiser une surface 3-D."

L'item du menu reprend le nom identifiant de la surface 3-D".

On a alors les correspondances suivantes :

- . l'action d'interaction : "sélectionner un élément de la BD, ici une surface 3-D, par le biais d'un "menu".
- . le moyen technique d'input : "menu visualisé et souris avec son curseur".
- . la saisie de l'information nécessaire au programme d'application : "visualiser" et "nom de la surface 3-D".
- . la répercussion dans la BD : "lecture des données de la surface 3-D".

On observe que l'on aurait pu mettre en oeuvre l'action d'interaction par un autre moyen technique d'entrée. Par exemple, on aurait pu utiliser une touche de fonction pour entrer la commande "visualiser" et le clavier pour introduire le "nom de la surface".

Le choix du moyen technique d'entrée est fonction

- . du choix personnel de l'utilisateur pour une interaction optimale.
- . des possibilités disponibles sur le système.

On remarque enfin que ces différentes étapes sont évidemment réalisées sous la conduite du LCI.

Passons brièvement en revue les actions d'interactions possibles et leurs moyens techniques généralement associés :

- a) POSITIONNER : choisir un lieu de la surface de visualisation pour insérer un élément graphique ou repositionner un élément graphique existant sur la surface. La mise en oeuvre technique de cette interaction sera notamment la souris avec bouton ou encore quatre touches de clavier orientant le curseur dans les quatre directions (gauche, droite, haut, bas).
- b) SELECTIONNER : choisir un lieu de la surface de visualisation pour sélectionner un élément existant.

Cet élément graphique pourra être un symbole géométrique, un point ou une ligne, un item de menu de commande,...
Les moyens techniques d'input seront le crayon lumineux, la souris, le joystick,...

- c) PEINDRE : dessiner selon le mouvement du curseur, un trait de caractéristique ajustable sur le support graphique.
Les moyens techniques d'input seront souvent le crayon lumineux ou la souris.
- d) PARAMETRISER : transmettre des paramètres (données et/ou commandes) pour activer une séquence d'actions du LGI.
Cela se fera soit, en sélectionnant un paramètre visualisé à l'écran (action "SELECTIONNER") mais aussi par le biais d'une méthode d'introduction de paramètres, très variable.
On pourra ainsi utiliser le clavier, la table de digitalisation, la boîte à fonctions (avec boutons et/ou potentiomètres dont la signification du paramètre transmis est prédéfinie).

I.1.2 L'interface Homme - Machine d'un LGI.

L'interface Homme - Machine reprend l'ensemble des choix effectués pour concevoir la "relation" entre l'utilisateur et le programme qu'il exécute.

Cette relation peut être minimale pour un programme "batch", ou beaucoup plus importante pour un programme dit "interactif".

Les choix effectués peuvent être clairs et répondre à des objectifs précis, ou être brouillons sans objectifs sous-jacents.

Vu le fonctionnement d'un LGI, on se rend compte de l'importance de son interface. C'est la première particularité d'un LGI.

Malheureusement, peu d'intérêt a été affecté à la définition d'une stratégie d'interfaçage ainsi qu'à celle d'une stratégie de conception de l'interface. Pourtant, un interface médiocre sera responsable de la difficulté de compréhension des concepts de fonctionnement d'un logiciel interactif et aussi de la difficulté d'utilisation de ce logiciel.

" A good user interface makes the program not only easy to learn but also easier and more efficient to operate" confirme Newman et Sproull (1979, p13).

Nous n'allons pas ici décrire les choix possibles pour concevoir un interface tant ils sont variés et peu formalisables.

Ainsi, Newman et Sproull affirment que l'"User - interface design, more than only other aspect of computer graphics, remains as much an art as a science". (1979, p14).

L'expérience et les erreurs passées sont en fait les seules références en la matière.

Nous nous attacherons donc plutôt au paragraphe I.2.2.2, à mieux cerner une démarche possible de conception d'un interface.

I.1.3 Les conséquences de la diversité des moyens techniques d'entrée/sortie.

Au contraire de beaucoup de logiciels classiques, les moyens techniques d'entrée/sortie (E/S) sont fondamentaux dans le cadre des LGI, car ceux-ci s'appuient sur un dialogue permanent avec l'utilisateur et ce, par le biais de moyens d'interaction diversifiés. Cette diversité est grande (différents types de supports de visualisation, clavier, crayon lumineux, table de digitalisation, souris, boîte à fonctions, joystick,...) et constitue donc la deuxième particularité des LGI.

Mais comment peut-on dans ces conditions, assurer la portabilité des programmes d'application s'ils dépendent des moyens techniques d'E/S? On veut en fait comme le soulignent Newman et Sproull (1979,p12) "(...) a set of subroutines that provides high level access to the graphics input-output hardware".

Pour faire face à ce défi, on présente la notion de "graphics package": "A good graphics package simplifies the programmer's task and makes it possible to write portable programs that can be run on different computers and with different displays".(Newman et Sproull,1979,p12).

Par l'usage de ce "graphics package" qui reprend ce qu'on appellera un ensemble de "primitives graphiques", on peut mettre à jour le modèle de processus de visualisation de la figure 4 (figure 9)

On considérera alors un module "Modelling system" utile à la transformation des structures de données de l'application tout en restant dans l'espace réel des éléments à visualiser.

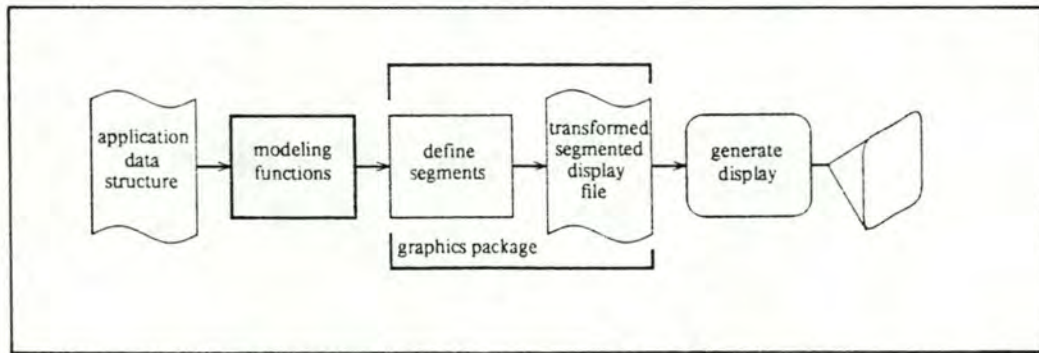


FIGURE 9: Output-process avec utilisation d'un graphics package.

(D'après Newman et Sproull, 1979, p433)

L'existence de ces graphics package ou logiciels graphiques de base (LGDB), dont l'utilisation se généralise, améliorera non seulement notre programmation par ses primitives, mais aussi l'indépendance du LGI par rapport aux moyens techniques d'E/S.

Soulignons tout de suite que la portabilité des LGI se généralisera lors de la réalisation d'un LGDB STANDARD.

Nous ne sommes à l'heure actuelle qu'au stade des projets de standardisation (GSPC d'ACM (SIGGRAPH GSPC, 1977), GKS (FNI, 1979), ...).

I.1.4 Conclusion.

Nous avons présenté rapidement et de façon succincte les caractéristiques d'un LGI.

On observe l'importance de l'interfaçage dont cependant la stratégie est difficilement formalisable. Notre démarche devra en tenir compte.

On remarque également la nécessité d'une discrimination des fonctions de haut niveau proche de l'application par rapport aux fonctions de bas niveau nécessaires à la conduite des moyens techniques d'E/S; ceci, afin de rendre les LGI portables.

Le LGDB constitue alors le "tampon" nécessaire.

Même en l'absence d'une standardisation des LGDB, on ne peut douter de l'intérêt de respecter dans notre démarche de conception, une modularisation des problèmes propice à la portabilité du produit final.

I . 2 Méthodologie de Conception : Premières Approches.

Quelles sont dans la littérature, les démarches de conception utilisées habituellement pour un logiciel de taille importante et quelles sont les démarches propres aux LGI?
Telles sont les deux questions auxquelles nous allons essayer de répondre.

Il est clair que nous ne développerons que la philosophie de ces différentes démarches.
Notre but est en effet de dégager leurs intérêts par rapport à une démarche possible de conception d'un LGI.
Dans la seconde partie du mémoire, la mise en oeuvre de chaque étape de conception sera précédée d'un rappel méthodologique précis.

I.2.1 Vers une démarche classique de développement de logiciels.

Si nous préconisons une méthodologie de conception applicable aux logiciels importants (systèmes d'information), nous pouvons énoncer le "méta-algorithme" (figure 10) issu des notions présentées par Messieurs Bodart et Van Lansweerde dans respectivement leurs cours d'Analyse fonctionnelle (Bodart, 1983) et de Méthodologie de développement de logiciels (Van Lansweerde, 1983).

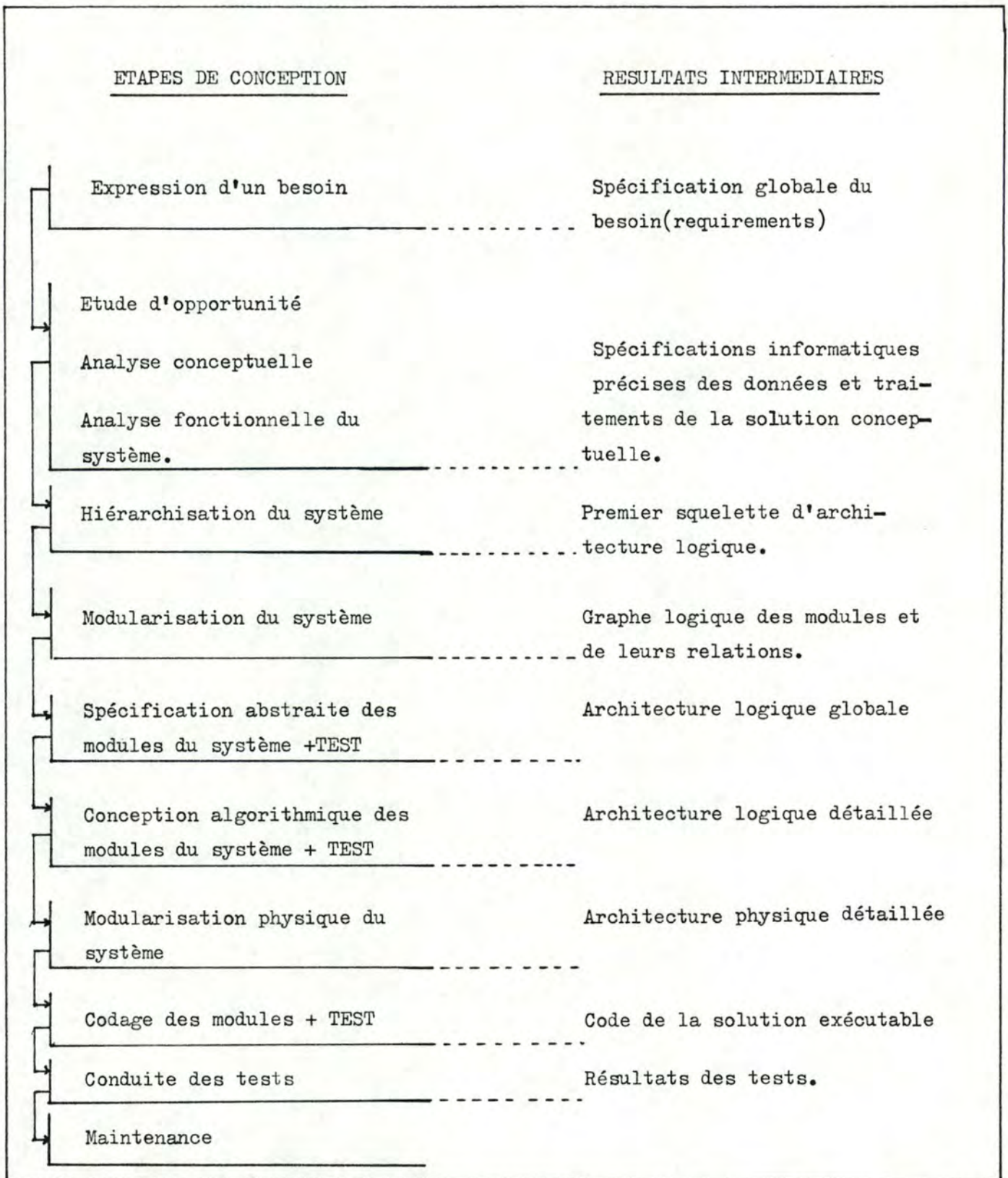


FIGURE 10 : Un Méta-algorithme de conception de logiciel.

La figure 11 permet de préciser un modèle de hiérarchisation suivant la hiérarchie "UTILISE" qui n'est d'ailleurs pas la seule relation envisageable.

<u>NIVEAUX</u>	<u>TYPES DE MODULES</u>
6	Composants fonctionnels issus des fonctions de l'analyse fonctionnelle
5	Composants fonctionnels de base
4	Composants d'E/S
3	Composants de contrôle
3-2	Composants d'attente
2	Composants "outils" existant déjà sous la forme de primitives formées par l'O.S, le SGBD,...
1	Composants de l'O.S

FIGURE 11 : Hiérarchisation type d'un système (avec relation "UTILISE")

A ce stade, il est bon de rappeler la notion de hiérarchie "UTILISE".

Soient deux modules A et B.

A utilise B si le fonctionnement correct de A dépend de la disponibilité d'une version correcte de B.

On concevra donc A en faisant abstraction de B sauf évidemment de ses spécifications.

Chacune des étapes de la démarche classique de la figure 10 fait

appel à des outils (modèles, langage de spécification, langage de programmation, méthodes de tests,...).

Nous détaillerons également ces outils lors de leur mise en oeuvre dans la seconde partie.

Attachons-nous à dégager les avantages de cette démarche classique.

Pour formaliser les avantages de la démarche classique, réexaminons les trois objectifs d'une démarche :

a) permettre la production de bons logiciels (voir les qualités attendues au paragraphe d'introduction de ce chapitre).

Cela est certainement possible. "Possible" car le concepteur n'est pas à l'abri d'erreurs de raisonnement, qui n'assurent pas par exemple la robustesse attendue du logiciel.

La production de bons logiciels est sans doute issue du haut degré de formalisme atteint dans cette démarche.

Que ce soit au niveau de l'analyse conceptuelle ou de design, chaque étape de la démarche a ses règles et principes, mais aussi ses outils de formalisation des résultats de l'étape.

Ainsi, la spécification des modules de l'architecture logique se fera par spécification dite "PRE-POST" afin d'éviter les spécifications en langage naturel moins précises.

Ce formalisme permettra ainsi, avec une hiérarchie "UTILISE",

- . de produire une maquette de la solution logique en fin d'analyse fonctionnelle.
- . de tester de façon systématique les résultats de certaines étapes du design (spécifications abstraites, algorithmes, codes).
- . de réduire les erreurs d'interprétation entre les différents concepteurs.
- . de simplifier la spécification et la conception des modules et d'en éliminer la redondance fonctionnelle.
- . de valider et maintenir le logiciel plus facilement.

Les deux dernières facilités sont directement issues de la hiérarchisation avec la relation "UTILISE".

C'est pourquoi cette relation est préférée.

On ne peut en effet oublier par exemple l'importance d'une maintenance aisée dont les coûts sont très importants (67% des coûts de logiciel en général).

b) être adaptée au type de logiciel :

La démarche classique est adaptée aux logiciels importants et notamment les systèmes d'information.

Vis-à-vis des LGI, on ne retrouve pas par contre leurs deux particularités (importance du design de l'interface, importance de la diversité des moyens techniques d'E/S), formellement exprimées dans une des étapes de la démarche.

Nous verrons au paragraphe I.3 comment on peut les intégrer.

Il y a donc une adaptation à réaliser à ce niveau, de la démarche classique.

c) être opérationnelle :

La démarche classique l'est.

I.2.2 Les démarches de conception liées aux LGI.

D'après nos recherches, il n'existe pas dans la littérature une démarche spécifique à la conception d'un LGI.

Par contre, et cela est essentiel, certains auteurs ont proposé des réponses aux deux particularités de tout LGI.

Ceci, dans un but d'une part, de bien spécifier l'interface Homme-Machine, d'autre part, d'assurer la portabilité des programmes d'application.

I.2.2.1 La mise en oeuvre d'un logiciel graphique de base.

Comme esquissé au paragraphe I.1.3, Newman et Sproull (1979) et Lucas (1979) voient la nécessité d'une séparation des LGI au moins en trois modules: le Logiciel graphique de Base (LGDB), le logiciel de mise en forme et le programme d'application (figure 12).

De là est née la notion de modularisation du LGI.

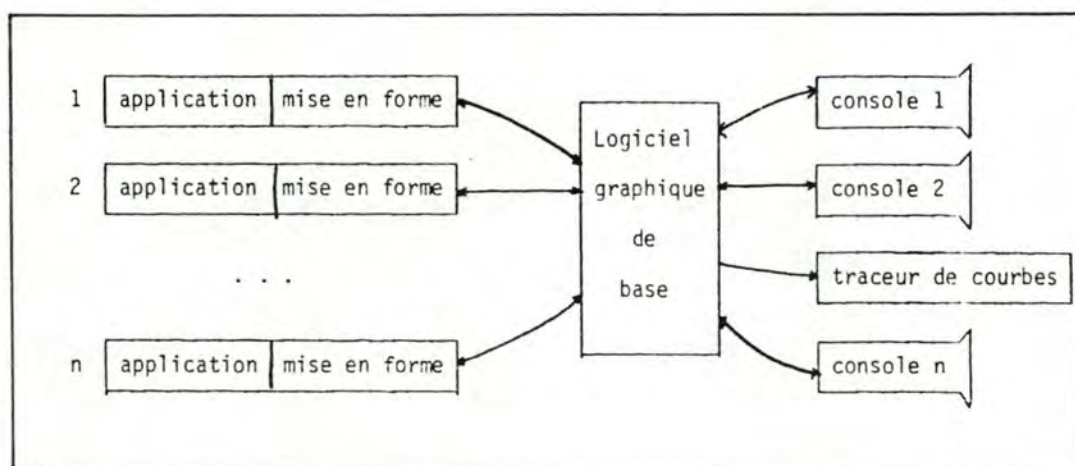


FIGURE 12 : Schéma de modularisation des LGI avec LGDB.

(D'après Lucas, 1982, p124)

I.2.2.1.1 Le logiciel de mise en forme.

Le rôle du logiciel de mise en forme est de permettre le passage des informations de la structure de données issues de l'application, aux informations graphiques et vice-versa.

Ce logiciel est dépendant de l'application et des informations qu'elles traitent.

Ce logiciel doit être conçu de telle façon que son interface avec le LGDB soit normalisée et permette donc la réalisation d'un LGDB commun à toutes les applications.

Nous pouvons préciser ce que serait ce logiciel de mise en forme. On pourrait ainsi, lui aussi, le subdiviser en plusieurs modules :

- . un module de transformation des données à visualiser pour produire une vue dans l'espace réel.

- . un module d'entrée regroupant les fonctions d'entrée de haut niveau (la reconnaissance de caractères introduits au crayon lumineux serait de ce type)

- . un module de génération de symbole prédéfini.

- . un module de génération de textes de haute qualité

etc...

I.2.2.1.2 Le logiciel graphique de base.

Le rôle du LGDB est "d'afficher sur l'écran une représentation des informations qui lui sont transmises et de recueillir les données graphiques destinées au programme d'application via le logiciel de mise en forme". (Lucas, 1982, p123). Il doit aussi assurer la portabilité des programmes d'application.

Ce logiciel doit donc être indépendant de l'application et assurer l'interfaçage entre les applications et les moyens techniques d'E/S.

Ce logiciel doit être conçu de telle sorte à assurer :

- . sa compatibilité avec tous les langages de programmation d'application. On ne peut adapter l'application et son langage au logiciel graphique de base.

- . son indépendance vis-à-vis des applications.
- . son indépendance vis-à-vis des moyens techniques d'E/S.

- . son indépendance vis-à-vis du système d'exploitation.

Comme annoncé au paragraphe I.1.3, le LGDB sera composé d'un ensemble de primitives graphiques. Newman et Sproull (1979, p439-440) expriment les fonctions principales d'un LGDB :

- . PRIMITIVES de BASE : celles-ci regroupent les primitives pour dessiner les points, lignes et textes simples (exemples : MOVETO (x,y); LINETO (x,y); DRAWTEXT (s))

- . PRIMITIVES de TRANSFORMATION : celles-ci regroupent les primitives de spécification de la fenêtre (window) et du champ (viewport) de visualisation à l'écran.

La figure 13 décrit les notions de fenêtre et de champ.

On ne se situe plus ici dans l'espace réel (mise en forme) mais dans l'espace écran.

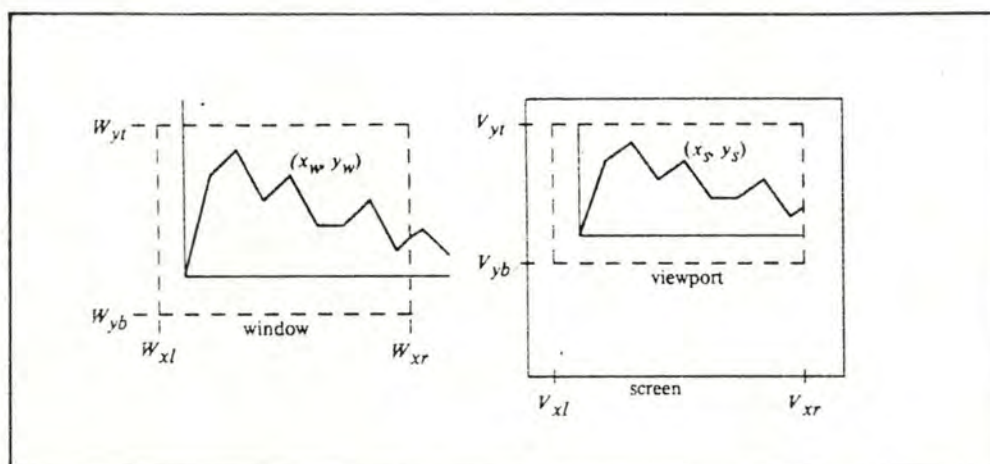


FIGURE 13 : Notion de fenêtre et champ de visualisation : schéma.

(D'après Newman et Sproull, 1979, p75)

. PRIMITIVES de SEGMENTATION : celles-ci regroupent les primitives de manipulation des segments graphiques.

Un segment est un ensemble de primitives graphiques associées logiquement et formant une unité.

Ainsi, si un segment est associé à un objet, on pourra visualiser, faire disparaître l'objet en modifiant les caractéristiques du segment (son caractère de visibilité).

. PRIMITIVES d'ENTREE : celles-ci regroupent les primitives d'activation et de désactivation d'un moyen technique d'Entrée.

. PRIMITIVES de CONTROLE : celles-ci regroupent les primitives de gestion des moyens techniques d'E/S (état des périphériques, modification de layout de l'écran,...).

I.2.2.1.3 Conclusion.

Afin d'assumer la diversité des moyens techniques d'E/S et de permettre la portabilité des programmes d'application, la modularisation des LGI est prônée.

Le pilier de cette modularisation est la spécification d'un LGDB.

On l'a déjà entr'aperçu, l'évolution est maintenant à la standardisation de ce LGDB. Nous devons en tenir compte.

"Comment intégrer cette approche modulaire dans notre démarche de conception?" sera une question à laquelle il faudra donc apporter une réponse positive.

4. SPECIFICATION de l'"INFORMATION DISPLAY": On spécifiera dans cette étape les caractéristiques générales des zones de l'écran (layout) ainsi que celles des objets visualisés.

Cette séparation permet de subdiviser les problèmes mais "The separation is not a clean one, however, and it is therefore not possible to design any one component independtly of the rest" soulignent Newman et Sproull (1979,p446).

On doit donc établir des choix précis pour chaque composant mais en les confrontant avec ceux établis pour les autres.

Ensuite, Green (1981) propose une démarche très proche de celle de Newman et Sproull.

La différence majeure avec cette dernière est la proposition d'un design en trois étapes :

1. une description de l'user's model comparable aux deux premières étapes de Newman et Sproull. La figure 15 illustre les correspondances entre les deux démarches à ce niveau.

Les correspondances indiquent l'identité sémantique des étapes des démarches.

2. une spécification formelle, basée sur la description de l'user's model, de l'interface.

3. une évaluation de cette spécification.

<u>GREEN</u>	<u>NEWMAN et SPROULL</u>
1 USER'S MODEL	
. TASK MODEL	----- a) USER'S MODEL
. CONTROL MODEL	----- b) COMMAND LANGUAGE
	c) INFORMATION DISPLAY
	d) FEEDBACK
2 SPECIFICATION de l'INTERFACE	
3 EVALUATION de la SPECIFICATION	

FIGURE 15 : Comparaison entre les méthodes de Green et Newman et Sproull.

(D'après Newman et Sproull, 1979 et Green, 1981)

On voit donc l'intérêt de la démarche de Green par l'utilisation d'un langage formel de spécification de l'interface.

Pour les besoins de cette formalisation, Green introduit, en plus des notions de "FUNCTIONAL ABSTRACTION" et de "DATA ABSTRACTION" traditionnelles, celle de "USER ABSTRACTION".

Si l'abstraction "FONCTION" tend à décrire les actions réalisées par un programme, si l'abstraction "DONNEE" décrit les opérations réalisables sur des données (Types abstraits), l'abstraction "UTILISATEUR", elle, décrira l'interaction de l'utilisateur avec le programme. Green (1981 a) présente de façon détaillée son langage de spécification de façon précise.

De surcroît, cette spécification est évaluée dans le but d'une vérification de l'adéquation de la spécification de l'interface avec la description de l'"USER'S MODEL".

En conclusion, cette démarche se veut plus formelle et plus sûre par l'évaluation des spécifications produites. Par contre, elle oublie les aspects c) et d) présentés par Newman et Sproull.

"Comment intégrer la conception de l'interface, élément essentiel d'un LGI, dans notre démarche globale de conception d'un LGI?" sera également une question importante.

I.3 Une démarche de conception d'un LGI: Notre choix.

I.3.1 Les critères de choix.

Tout choix doit se faire selon un ou plusieurs critères définis à priori. Il nous semble peu orthodoxe de fixer des critères à posteriori: ils risquent d'être dépendants d'un choix, peut-être même prédéfinis.

Alors, à quoi bon choisir, si cela est déjà fait au départ ou si les critères sont biaisés.

Nous avons exprimé les qualités attendues d'une démarche de conception :

- . Être opérationnelle,
- . Être garante des qualités d'un logiciel,
- . Être en accord avec les particularités d'un LGI (importance de l'interface, diversité des moyens techniques d'E/S).

Nous proposons donc le critère de maximisation de ces trois qualités de la démarche.

I.3.2 Le choix.

Le critère de choix nous amène tout d'abord à suivre la démarche classique (figure 10) dont l'intérêt principal est, dans la mesure où sa mise en oeuvre est de qualité, la garantie des qualités d'un logiciel.

Nous devons en surcroît nous efforcer d'intégrer les principes de conception liés à la modularisation nécessaire d'un LGI (voir paragraphe I.2.2.1) et au design de l'interface Homme-Machine (voir paragraphe I.2.2.2).

Tout d'abord, concernant la modularisation nécessaire d'un LGI, remarquons que la hiérarchisation du système est propice à l'introduction d'un LGDB dont les primitives seraient utilisées par les modules de niveau supérieur.

Conformément à la figure 11, nous proposons de placer

- . au niveau 2, un LGDB dont nous définirons les caractéristiques au chapitre VI.
- . au niveau 4, les modules d'E/S visibles par des primitives de haut niveau.
- . au niveau 6, les modules assurant les fonctions de l'analyse fonctionnelle.

Nous préciserons les principes de hiérarchisation et de modularisation, dans le cadre du LGI développé en seconde partie, au chapitre IV.

Ensuite, concernant le design de l'interface, nous proposons, comme le préconise Green, d'inclure une spécification abstraite de l'interface à partir de modules de l'architecture logique. Cette spécification abstraite reprendra (figure 15) :

- . le task model ou l'user model
- . le control model ou le command language
- . l'information display
- . les feedbacks.

Malheureusement, nous ne pourrions spécifier de façon formelle, par le langage prôné par Green, l'interface (étape 2 de la méthode de Green).

En effet, nous n'avons pu disposer de la référence (Green, 1981 a) nécessaire pour prendre connaissance de ce langage.

Nous nous reporterons donc sur un langage plus informel.

I.4 Conclusion.

Après avoir énoncé les caractéristiques principales d'un LGI, ce chapitre fut l'occasion d'ébaucher une démarche de conception de LGI. Ce domaine ne faisant pas l'objet de recherches significatives dans la littérature, il nous a donc semblé intéressant de proposer une démarche.

Cette démarche est proche de la démarche "classique" (Bodart et Van Lansweerde), où la modularisation tient compte de la notion de LGDB et de portabilité du LGI par rapport au moyen d'E/S, et où aussi la spécification des modules est enrichie d'une spécification de l'interface.

Dans la seconde partie de ce mémoire, comme annoncé, nous appliquerons cette démarche à la conception d'un logiciel de manipulation interactive de surface 3-D.

Ainsi, au chapitre II, nous présenterons les spécifications globales du LGI issues de l'expression des besoins, des objectifs du LGI.

au chapitre III, nous réaliserons l'analyse conceptuelle des fonctions du LGI.

au chapitre IV, nous réaliserons les étapes d'hierarchisation, de modularisation et de spécification abstraite du système et présenterons les choix sous-jacents à ces 3 étapes.

au chapitre V, nous compléterons les spécifications abstraites des modules par la spécification de l'interface.

au chapitre VI, nous réaliserons les étapes de conception algorithmique, de modularisation physique et de codage du système et présenterons les choix sous-jacents à ces 3 étapes.

et enfin , nous présenterons et réaliserons les plans de tests aux différents niveaux possibles (spécifications abstraites, algorithmes, code).

Cette découpe des étapes de la figure 10 au sein des chapitres du mémoire répond à quatre principes :

- Séparation des aspects conceptuels des autres aspects logiques et physiques (chapitres II,III).
- Séparation des aspects "spécification" et "mise en oeuvre" (chapitres IV,V,VI).
- Mise en évidence de la spécification de l'interface (chapitre V).
- Importance de l'aspect "évaluation" du produit.

Chaque étape fera l'objet d'un rappel des outils et méthodes nécessaires à sa réalisation.

Voilà donc établi notre plan de travail!

IIème PARTIE
UN LOGICIEL GRAPHIQUE INTERACTIF :
MANIPULATION INTERACTIVE DE SURFACES 3-D .

CHAPITRE II

DU PROJET-CADRE AUX SPECIFICATIONS GLOBALES DU LOGICIEL

II . 1 De l'expression d'un besoin aux objectifs du logiciel.

Avant de préciser le projet-cadre qui reprend l'ensemble des objectifs du logiciel, nous allons énoncer les besoins qui en furent à la base.

Après l'élaboration de ce projet-cadre, nous détaillerons pour celui-ci les spécifications globales, encore appelées "requirements", précisant ce que l'utilisateur attend du logiciel.

Enfin, nous développerons pour deux applications (géographique et chimique), leurs besoins et leurs objectifs propres. Nous entendons par application ici, le fait que la surface tri-dimensionnelle (3-D) représente un phénomène géographique ou chimique.

II.1.1 Les besoins.

Sous l'impulsion du Docteur André Michel, membre du département de chimie des Facultés de Namur, nous avons émis l'idée d'un outil d'étude des surfaces 3-D, et ceci, par la puissance de visualisation et de manipulation de ces surfaces offerte par les outils graphiques. Ces surfaces 3-D peuvent être issues de phénomènes spatiaux observables ou de la création libre d'un utilisateur. Nous préciserons une classification des surfaces 3-D au paragraphe II.2.1.1.

Au delà de ce premier besoin, nous avons pressenti la nécessité d'une ouverture très large, en terme de ses applications, de cet outil graphique d'étude des surfaces 3-D. En effet, les formations de chimiste et de géographe dont le Docteur Michel et moi-même étions titulaires nous avaient conforté dans l'idée de l'utilité de ce genre d'outils à la fois puissants mais aussi adaptés aux besoins d'applications spécifiques et diverses.

Soulignons de suite que ce genre d'outils n'existe pas à notre connaissance, tout au moins en Belgique.

Les besoins exprimés nous semblent donc réels et consistants.

II.1.2 Les objectifs du logiciel:Le projet-cadre de notre étude.

De cette analyse des besoins, nous nous sommes donc fixés comme objectif de réaliser un logiciel graphique de saisie, de visualisation et de manipulation de surfaces issues de phénomènes 3-D quelconques.

Ce logiciel se doit, en plus de comporter les qualités indispensables de tout logiciel (voir introduction du chapitre I), de permettre une exécution interactive aisée aux utilisateurs non avertis en informatique.

On remarque que l'énoncé de ces objectifs vise à la fois:

- . les fonctions du logiciel (saisie, visualisation, manipulation de surfaces 3-D quelconques).
- . les moyens utilisés pour ce logiciel (les outils graphiques).
- . les qualités attendues du logiciel (qualités classiques ainsi que le caractère interactif).
- . les utilisateurs du logiciel (non expérimentés en informatique et variés par la diversité des surfaces 3-D étudiables).

Ces objectifs sont ambitieux quant à leur complexité, à l'investissement "temps" nécessaire et enfin, au souci de généralisation des fonctions du logiciel.

De plus, l'outil graphique et la qualité d'interactivité ne sont pas aisés à mettre en oeuvre en regard des logiciels classiques.

Nous avons défini de tels objectifs afin de contraindre notre travail dans une perspective d'ouverture à une évolution possible du logiciel.

Mais nous devons restreindre, pour ce mémoire, la portée de certains objectifs; nous préciserons cela plus loin.

II.1.3 La justification de ces objectifs.

Les objectifs se doivent d'être justifiés de manière précise afin d'éviter de s'en fixer d'inutiles ou d'impossibles.

Les objectifs "fonctionnels" (saisie, visualisation, manipulation de surfaces 3-D quelconques) sont issus de l'expression même des besoins d'un outil d'étude des surfaces 3-D, étude qui nécessitera la saisie, la visualisation et la manipulation de la surface.

Les objectifs "utilisateurs" sont également issus des besoins exprimés en II.1.1.

En effet, l'ouverture à des applications diverses (il n'y a pas de restrictions sur les phénomènes 3-D à étudier) implique que les utilisateurs non-informaticiens peuvent aussi avoir accès à ce genre d'outils.

Les objectifs "qualités" se déduisent eux intuitivement. Précisons simplement que l'interactivité est essentielle à la fois pour faciliter l'étude des phénomènes mais aussi pour améliorer l'accès du logiciel aux novices.

Remarquons également que la qualité d'extensibilité est importante au vu de la volonté exprimée, d'ouverture du logiciel aux applications diverses.

Enfin, les objectifs "moyens" proviennent également des besoins exprimés mais constituent aussi une nécessité si l'on veut visualiser et manipuler des surfaces 3-D dans de bonnes conditions.

II . 2 Cadre général du projet informatique.

Ce cadre général nous permettra de préciser les objectifs du projet-cadre (paragraphe II.1.2) afin d'énoncer ensuite les spécifications globales du LGI.

Pour cela, passons en revue les différents objectifs énoncés.

II.2.1 Par rapport aux objectifs fonctionnels.

On veut saisir, visualiser et manipuler une surface 3-D quelconque.

Nous devons spécifier le type de surfaces que l'on étudiera.

Ensuite, nous préciserons les fonctions importantes attendues du logiciel.

II.2.1.1 Les différents types de surfaces 3-D et notre choix.

On peut proposer la classification générale des surfaces 3-D de la figure 16.

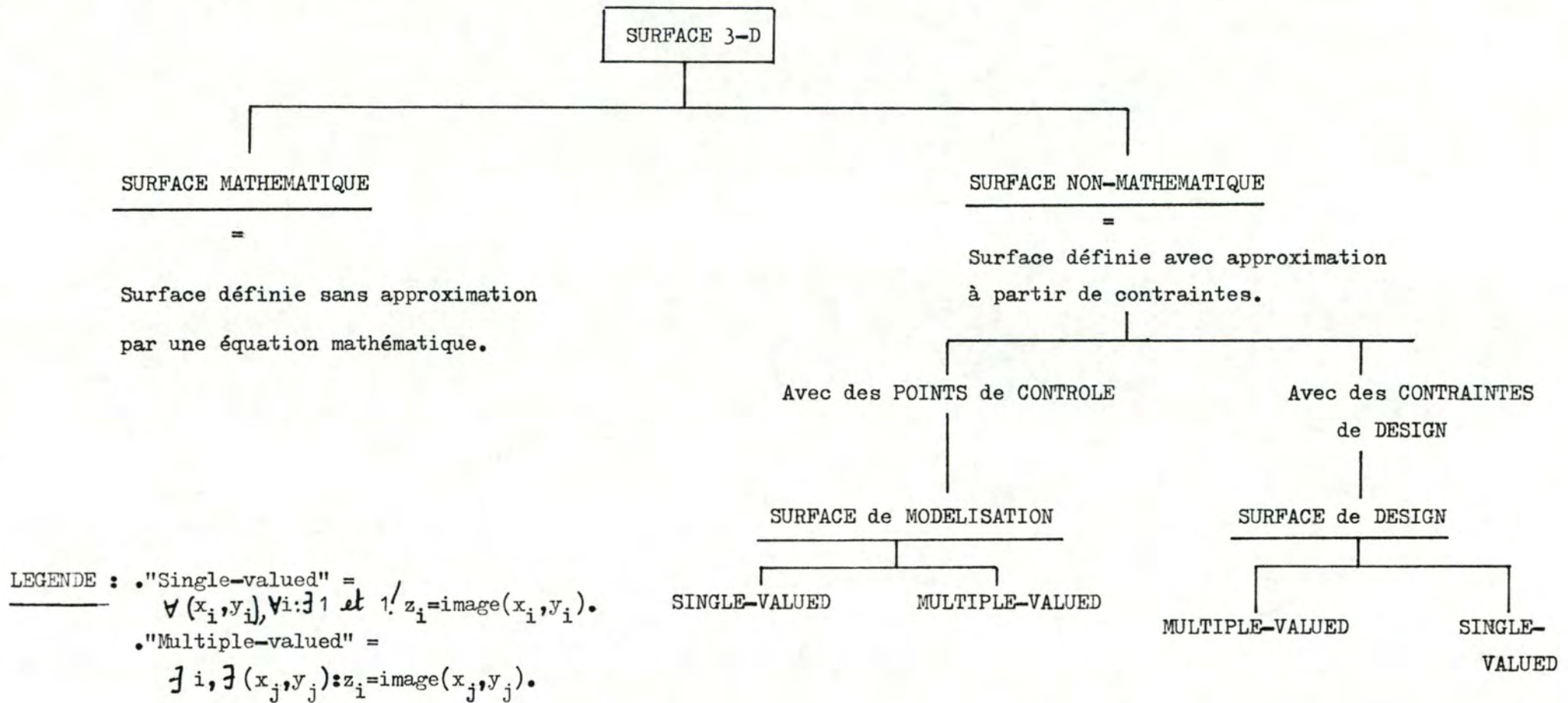


FIGURE 16 : Classification des Surfaces 3-D.

Les principes de celle-ci sont le type de définition de la surface (équation mathématique, points de contrôle, contraintes de design) et l'existence de points multiples (multiple-valeur, simple-valeur) pour un même couple de coordonnées.

Cette classification nous semble indispensable car selon le type de surface, sa représentation en machine, sa visualisation et sa manipulation sont différentes. De plus, cette classification nous permet de choisir les surfaces qui seront concernées, au départ, par le logiciel à concevoir. Ce choix est nécessaire. En effet, le temps et les moyens disponibles nous empêchent d'envisager une implémentation globale. Ce choix de surfaces constitue donc la première restriction de notre travail. Il sera plus raisonnable de réfléchir à la possibilité d'extension (voir paragraphe II.2.3) du logiciel envers une large variété de surfaces.

Notre choix s'est porté sur les surfaces issues d'un phénomène observé, où sont donc exclues les surfaces de design définies par des équations ou des points de contrôle. On étudiera donc les surfaces dites de "modélisation" (simple et multiple valeurs) et mathématiques.

Ce choix est principalement guidé par la saut de complexité qu'implique les surfaces de design, notamment au niveau de leur saisie et de leur représentation en machine. Dans un premier temps, nous n'avons pas voulu franchir ce pas.

II.2.1.2 Les grandes fonctions du logiciel.

Les grandes fonctions du logiciel peuvent être identifiées par la simple logique. Celle-ci conduit à la découpe fonctionnelle de la figure I7.

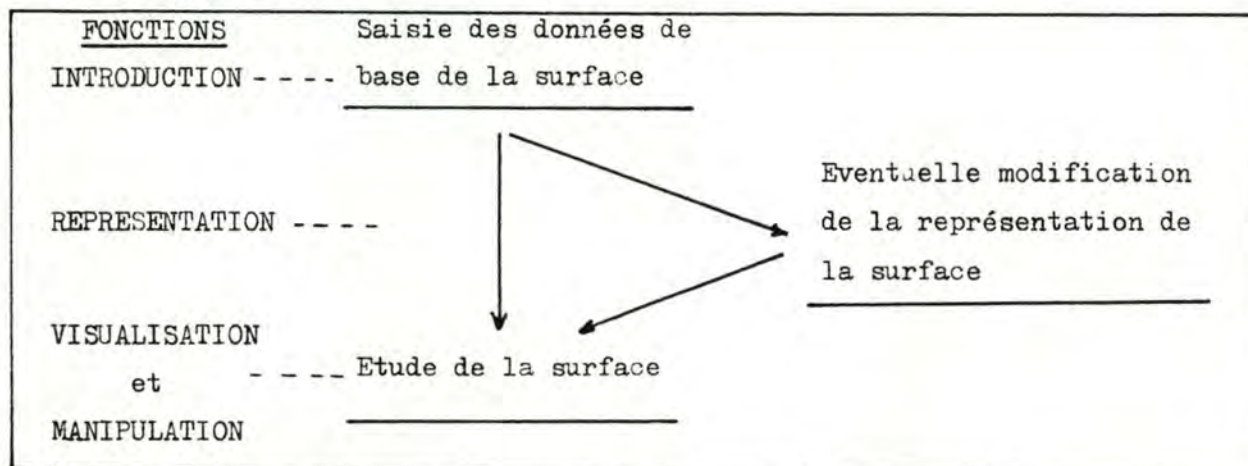


Fig.I7: Les fonctions de base du logiciel général.

On distingue (fig.I7) les quatre fonctions:

- . INTRODUCTION : selon notre choix de surfaces, on introduit la surface 3-D par le biais d'équations ou de points de contrôle. A ce moment, la surface aura sa représentation sous forme de coefficients d'équations ou d'une liste de points.
- . REPRESENTATION : Cette fonction permet de modifier une représentation inadéquate pour la manipulation ou la visualisation de la surface.
- . VISUALISATION : A partir d'un état i issu de manipulations antérieures, on construit ici les données nécessaires à la visualisation de la surface dans cet état i .
- . MANIPULATION : Cette fonction assure le passage de la surface de l'état i à l'état $i + 1$ par la mise en oeuvre d'une manipulation.

La notion d'ETUDE de la surface regroupe les fonctions de manipulation et de visualisation afin de signifier le "dialogue" permanent entre la mise en oeuvre d'une manipulation et la visualisation du résultat de cette mise en oeuvre.

Détaillons à présent les spécifications globales concernant ces quatre fonctions.

II.2.1.2.1 Introduction de la surface.

Cette mise en ordinateur sera évidemment variable suivant la surface que l'on désire étudier.

Remarquons également que cette phase risque, rapidement d'être laborieuse si l'on doit introduire un grand nombre de points de contrôle. Selon MORVAN (1976, P.169), cette étape d'introduction reste l'un des plus gros problèmes de la graphique de la représentation de surfaces 3-D complexes. Soyons-y attentifs.

a. Surface Mathématique

Deux types d'équations sont envisageables:

- équation algébrique (Simple valeur): $Z = f(x, y)$ ou $F(x, y, z) = 0$.
- équation paramétrique (quelconque) : $X = f(u, v)$
 $Y = g(u, v)$
 $Z = h(u, v)$.

Cette situation est triviale: on introduira les coefficients de l'équation puis on les validera.

Cette notion de validation est essentielle pour la fonction d'introduction des données de la surface.

b. Surface de Modélisation.

On se devra ici de saisir la liste des points de contrôle à partir de laquelle l'étude sera conduite. Ces points se caractériseront par des triplets de coordonnées 3-D (x_i, y_i, z_i ; $i: 1, \dots, p$) répartis de façon aléatoire dans l'espace. Pour fixer le cadre dans lequel se situent les coordonnées, il sera bon de signaler la signification des trois dimensions.

Plusieurs possibilités sont envisageables pour ce qui est de l'introduction des coordonnées 3-D :

- . Entrée au clavier des valeurs. Cette procédure est peu élégante quand le nombre de points s'accroît.

- . Entrée par plans successifs. Dans un plan de valeur $X = x_i$ ou $Y = y_i$ ou $Z = z_i$, l'utilisateur en fonction d'un système de valeurs de coordonnées choisies, introduira des points (x_i, y, z) ou (x, y_i, z) ou (x, y, z_i).

Une succession de plans adéquatement définis permettrait l'introduction globale des (x, y, z). Cette méthode me semble opportune quand l'utilisateur dispose de coupes dans sa surface. WU (1977) a implémenté une telle pratique.

Cette introduction par plans successifs peut se faire par définition du système d'axes dans le plan de l'écran ou à la table de digitalisation; le curseur pouvant alors indiquer le lieu dans le "plan d'introduction", le clavier lui, permettant l'entrée de la 3ème valeur de coordonnée.

Aucune de ces méthodes n'est idéale mais semblent les plus courantes. On les implémentera toutes les deux.

Remarquons dès à présent que la seconde méthode permet l'introduction de surfaces complexes définies par les courbes quelconques dans les plans de coupe (WU, 1977). Cette méthode est donc la porte ouverte à la saisie de surface de modélisation multiple-valeur ou même de design.

Il nous reste à préciser des spécifications complémentaires:

- l'utilisateur aura le choix par menu, de la méthode d'introduction.
- l'utilisateur pourra après avoir entamé ou même terminé la saisie de données, revenir au point de départ. Par contre si il est satisfait d'une saisie correcte, il la validera.

- Tout point de contrôle introduit lors de la saisie pourra être corrigé.
- l'utilisateur aura à sa disposition après validation, la possibilité de lister sur papier l'introduction des points
- La validation entraînera
 - la possibilité de passer à la phase logique suivante
 - le stockage séquentiel des triplets de coordonnées.
- l'utilisateur aura à sa disposition après validation d'une saisie, la possibilité de consulter et de mettre à jour le fichier s'y rapportant.
- la non-validation sera sans effet sur le déroulement du logiciel : on restera à la phase de saisie.

II.2.1.2.2. Représentation de la surface.

Cette phase logique ne concerne pas directement l'étude de la surface. Mais la façon dont celle-ci est représentée en machine va conditionner les étapes importantes de manipulation et de visualisation graphiques.

Plusieurs méthodes existent. On peut présenter les critères suivants pour choisir une représentation :

1. Le type de surface introduit.
2. Le degré de compacité.
3. Le degré d'accessibilité d'un élément de la surface.
4. Le degré de manipulation permis.
5. La facilité de visualisation permise

De la littérature ressort la classification suivante :

- A Une liste des points de contrôle.
- B Une équation de la surface.
- C Une matrice provenant d'une interpolation.
- D Un ensemble de matrices provenant d'une approximation globale.
- E Un ensemble d'équations provenant d'une approximation.

a. Une liste des points de contrôle.

Cette représentation ne demande aucune transformation à partir de la phase de saisie des données. Mais au vu des critères 2 et 3, il est évident que cette éventualité (surtout pour un nombre élevé de points) devient peu praticable.

b. Une équation de la surface.

Par rapport aux critères 2 à 4, cette représentation est idéale. Mais elle ne peut s'appliquer évidemment qu'aux surfaces mathématiques (critère 1.)

c. Une matrice.

L'élément matriciel serait z_i correspondant à la ligne i ($=x_i$) et à la colonne j ($=y_j$).

Cette représentation est en accord avec les surfaces simple-valeur seulement (critère 1).

Seules les surfaces de faible amplitude de valeur X et Y seront admises, sous peine d'une trop grande généralisation de la surface. Si l'accès à un élément est facile (critère 3), la maniabilité (critère 4) et la compacité (critère 2) d'une matrice ne sont pas toujours aisées. Comme nous le verrons plus loin, cette représentation se prête très bien à la visualisation (critère 5). L'attribution d'une valeur z_i à chaque élément de la matrice se fait par une méthode d'interpolation locale.

Nous proposons d'implémenter cette méthode de représentation matricielle si les méthodes plus complexes mais plus performantes décrites ci-dessous s'avèrent difficiles à implémenter.

d. Un ensemble de Matrices : La méthode de BRUNET (1983)

On réalise ici une approximation globale par la méthode des polynômes spline, mais cela à partir d'une répartition régulière (sur un maillage) des points de contrôle. Brunet décrit donc un algorithme qui à partir des points de contrôle aléatoires (dans leur répartition) estime des points réguliers (dans leur répartition). Ces points réguliers construiront les matrices d'approximation.

L'utilisation des splines implique :

- une continuité forte de la surface (à la dérivée seconde près).
- une très bonne proximité de l'approximation à la réalité.
- un calcul assez long des matrices d'approximation.
- un calcul rapide d'un point à la surface (critère 3).
- une grande facilité de transformation et visualisation
(critères 4 et 5)
- une bonne compacité (critère 2).
- une surface de modélisation simple-valeur (critère 1).

Cette méthode est donc assez séduisante pour ces dernières.

e. Un ensemble d'équations.

Cet ensemble (unitaire ou non) nécessite une approximation à partir d'un ensemble de points de contrôle répartis de façon aléatoire dans l'espace.

De façon générale, deux attitudes sont possibles (BRUNET 1983, p.104):

1. Approximation GLOBALE ; la Méthode des Moindres Carrés.

On ajuste ici par régression un polynôme de degré désiré pour approcher la complexité de la surface à modéliser. Mais on observe que cette technique devient peu fiable en termes de proximité de l'approximation à la réalité, quand celle-ci devient complexe.

2. Approximation LOCALE.

Dans ce domaine, deux techniques existent également:

a. Méthode d'AKIMA (1978) : Par une méthode de triangulation, l'algorithme proposé approxime dans chaque triangle un polynôme du 5ème degré mais ceci afin d'obtenir une continuité du premier ordre (à la dérivée première près) entre les triangles. Pour un nombre élevé de points de contrôle ($P \geq 30$), cette méthode semble insatisfaisante quant à la proximité de l'approximation à la réalité (BRUNET 1983, p.107)

b. Méthode dite "Paramétric Patches" : A partir de chaque portion de la surface (Patch) est approximée une équation paramétrique par la méthode des splines. L'utilisation des splines semble la plus efficace par

- sa simplicité algorithmique (exemple dans LUCAS 1982, p.223)
- sa rapidité de calcul car le degré des courbes d'approximation est au maximum de quatre (NEWMAN et SPROULL (1979, p.324))

FOLEY (1982) donne une explication claire du calcul. ROGERS (1976) en donne une implémentation possible. Seule pour nous reste obscure la méthode de subdivision en patches de l'ensemble des points de contrôle. Au vu des critères 2 à 5, elle est une bonne représentation. Elle a l'avantage d'accepter des surfaces multiple-valeur (critère 1)

En conclusion, nous proposons d'implémenter les méthodes de représentation suivantes:

- . pour les surfaces mathématiques, on représentera celles-ci par les coefficients de leurs équations.
- . pour les surfaces de modélisation simple-valeur, on implémentera la méthode de BRUNET et son approximation par spline. Nous avons d'ailleurs reçu de son auteur un exemple d'implémentation.
- . pour les surfaces de modélisation multiple-valeur, idéalement la méthode du "paramétric patches" est indiquée.

II.2.1.2.3. Visualisation de la Surface.

Cette phase logique reprend simplement les spécifications du "display" de la surface 3-D. Ce display se fera sur écran CRT type line-drawing".

La visualisation la plus esthétique nécessite certainement deux choses :

a. Une visualisation avec maillage "WIRE-FRAME" (NEWMAN et SPROULL, 1979, p.325.) C'est une technique où la surface est esquissée par un ensemble de courbes orthogonales pour des valeurs incrémentées, de u et v si on a une représentation paramétrique, de x et y si on a une représentation algébrique.

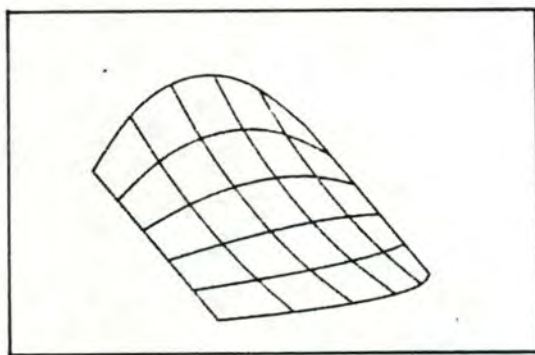


Fig I8: Visualisation en "Wire-frame".

b. Une visualisation de la profondeur. Ceci peut se faire de différentes manières. D'après NEWMAN (1979, p.298.) pour les objets représentés en WIRE-FRAME, la méthode d'élimination des lignes cachées est la plus adéquate. Et cela malgré qu'elle comporte de nombreux calculs.

Spécifions ici que l'élimination des parties cachées ne se fera que lorsque l'utilisateur le désire, et donc se fera à la demande.

L'utilisateur aura la possibilité d'orienter les axes x , y et z .

Par défaut, ils seront orientés de façon LEFT-HANDED (fig I9) par rapport à l'écran de visualisation.

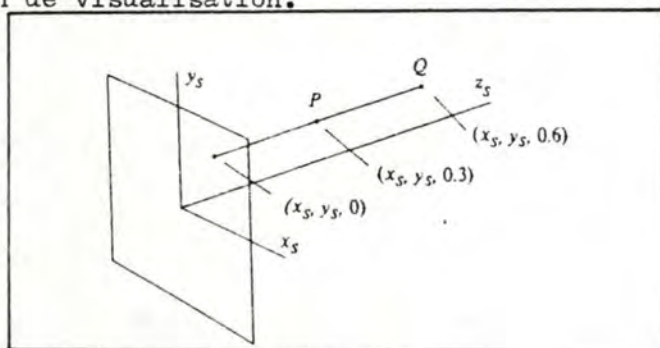


Fig. 19: Un système d'axes Left-handed.

L'utilisateur pourra visualiser les axes avec les unités de mesures du phénomène observé.

Toute surface visualisée devra subir les transformations nécessaires à sa visualisation et qui comprennent, dans l'ordre chronologique :

1. un ajustement à la fenêtre de visualisation (Window) dans l'espace-objet.
2. un ajustement au "point de vue" que l'on veut adopter par rapport à l'objet dans l'espace-objet.
3. un ajustement par élimination des parties externes à l'espace-écran (= clipping)

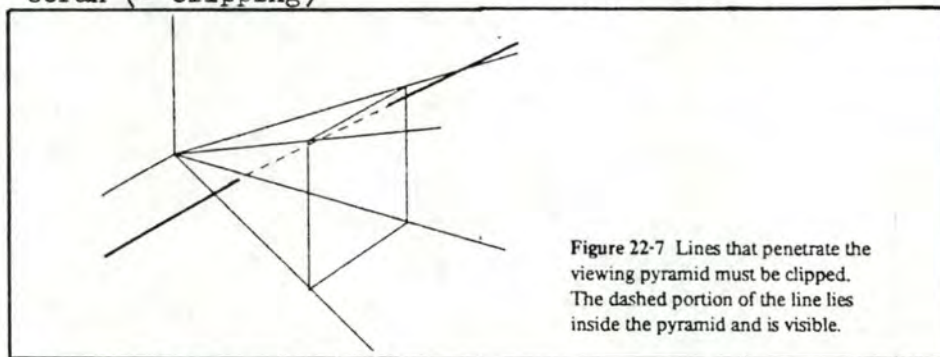


Fig. 20 : Le clipping.

4. un ajustement au champ de visualisation (View port) dans l'espace-écran.
5. un ajustement afin de visualiser la profondeur à l'écran (=transformation perspective).

On se rapportera au chapitre 22 de NEUWMAN et SPROULL (1979) pour plus de détails sur les transformations.

Rappelons que certaines de ces transformations pourront être réalisées par les primitives du LGDB. Ainsi on peut voir que les transformations n°1, 3 et 5 seraient réalisées par les primitives du LGDB décrit au chapitre I (paragraphe I.2.2.1.2.)

- . Toute surface visualisée pourra faire l'objet d'un tirage hard-copy.

- . Pour les surfaces simple-valeur, la surface pourra être présentée sous forme de bloc-diagramme. (c-à-d avec abaissement de perpendiculaires de la surface vers les axes (X,Y) ou (X,Z) ou (Y,Z))

II.2.1.2.4. Manipulation de la Surface.

Nous proposons les manipulations suivantes à réaliser par le logiciel.

A. Pour une surface.

- SCALING : Ceci correspond à un changement d'échelle de la surface selon un certain facteur et une certaine direction.
- ROTATION : ceci correspond à une rotation de la surface selon un certain angle et autour d'une certaine direction.
- TRANSLATION : ceci correspond à une translation de la surface suivant un certain axe et d'un certain intervalle.
- FENETRAGE : ceci correspond à une limitation de la surface visualisée du système de coordonnées réelles selon certaines valeurs de (x,y,z) min et (x,y,z) max.
- COUPE : Ceci correspond à couper la surface selon un certain plan et à visualiser le plan de coupe. La réalisation d'une coupe multiple reviendra alors à la visualisation d'isolignes issues chacune d'une coupe. Dans ce cas, les coupes seront parallèles et auront un certain intervalle.
Lors de la réalisation d'une coupe, on visualisera à la fois la surface de base et le plan de coupe. Ceci afin de mieux apprécier les caractéristiques de la coupe.
- ACCES A UN POINT : ceci correspond au pointage d'un lieu de la surface et au renvoi de la coordonnée de ce point (x_i, y_i, z_i)

B. Pour deux surfaces.

- COMPARAISON SPATIALE : Ceci correspond à la visualisation simultanée de deux surfaces, à leur addition, soustraction éventuelles. La visualisation simultanée implique qu'à tout moment le viewport pourra être divisé en deux parties.
- COMPARAISON STATISTIQUE : Ceci correspond à la mesure de l'écart moyen des valeurs entre les deux surfaces. Cette opération implique la restitution à l'écran d'éléments d'appréciation de la comparaison statistique.

Nous devons compléter les spécifications de manipulation par :
Trois opérations doivent à tout moment être possibles :

- la poursuite d'une manipulation avec une surface déjà transformée.

- le retour à la surface de base (avant la 1ère manipulation) et le retour à la surface précédant la dernière manipulation.
- la mémorisation d'une surface manipulée.

Il est clair que ces spécifications globales par rapport aux "objectifs fonctionnels" seront précisées et mise en oeuvre dans la suite du travail.

II.2.2. Par rapport aux objectifs " moyens ".

Nous avons souhaité utiliser un outil graphique pour la réalisation de ce logiciel.

Cet outil graphique devrait comprendre un écran de type line-drawing à haute résolution et un ensemble de moyens d'interaction (boîte de fonctions, potentiomètres, clavier, joystick, table de digitalisation, souris, ..)

Cet outil graphique doit être compatible avec un LGDB proche des projets de standardisation du GSPC ou du GKS.

Nous décrirons au chapitre VI, les implications de ces spécifications lors de la présentation de nos choix d'implémentation physique qui sera alors faite.

II.2.3. Par rapport aux objectifs "qualités "

Nous désirons un logiciel fiable, réutilisable, efficace, interactif, convivial, sûr, portable et dont la maintenance est aisée. Nous insistons ici sur les aspects :

- . de portabilité, vu la diversité des moyens techniques d'E/S.
- . d'extensibilité.

Selon notre optique, il est important de distinguer deux niveaux d'extensibilité :

a. Le logiciel doit pouvoir facilement accepter des modules de traitements complémentaires dans le cadre d'une application précise. Par exemple, pour l'étude des densités urbaines géographiques, il nous sera indispensable d'ajouter des modules d'analyse de régression, des résidus, ...

b. Le logiciel doit pouvoir facilement accepter des modules de traitements complémentaires dans le cadre d'une extension des possibilités du logiciel. Et cela en rapport avec :

- l'introduction d'un nouveau type de surface.
- l'introduction en conséquence de la représentation de ce nouveau type.
- l'ajout d'algorithmes de visualisation de la profondeur.
- l'ajout de manipulations de surfaces.

- d'interactivité.

Cette interactivité doit se retrouver à tous les niveaux du logiciel car

- . elle est un élément de notre objectif de base,
- . elle est une source d'enrichissement pour l'étude du phénomène manipulé.

Deux conceptions d'interactivité peuvent être envisagées :

a Libre choix : A chaque étape d'une session d'utilisation, l'utilisateur pourra choisir toutes les éventualités possibles même irrationnelles. Cette conception implique des jeux de tests de validité des choix effectués pour ne pas "emballer" le logiciel dans une voie aberrante.

b. Guidance : Sous cette forme, à chaque étape, l'utilisateur n'aura le choix que des options logiques au vu des décisions précédentes. Cette conception permet d'alléger la programmation des tests de validité et faciliter le cheminement de l'utilisateur.

Nous choisirions plutôt la seconde conception.

L'interactivité du logiciel se retrouvera principalement à deux niveaux (qui englobent la presque totalité du cheminement):

a L'introduction de données.

b L'introduction par "Menus" d'options concernant le type d'introduction de données,
les manipulations des surfaces,
la visualisation des surfaces,
la non-exécution d'un choix effectué, etc...

- d'efficacité, notamment quant à la possibilité de conserver les résultats d'une étude d'une surface.

II.2.4. Par rapport aux objectifs "utilisateurs "

L'utilisateur doit pouvoir comprendre et utiliser les concepts du logiciel de façon aisée.

L'utilisateur doit pouvoir comprendre et utiliser le langage de commandes du logiciel de façon aisée. Et ceci quel que soit le type d'application.

II.3. Cadre général de l'application géographique.

II.3.1. Les besoins liés à l'application géographique.

L'idée de cette application géographique a pour origine notre travail de fin de licence en géographie où un outil d'étude de la variation spatiale des densités de population (en milieu urbain) eut été très fructueux.

Le phénomène 3-D à étudier est donc la variation spatiale des densités de population mesurée par unités statistiques (communes avant et après fusion, secteurs statistiques)

En plus de cette étude des densités de population réelles, on était intéressé par celle de la variation modélisée des densités de population.

Cette modélisation se fait par le biais de modèles (à choisir), dont les paramètres sont estimés à partir de la variation réelle des densités.

Ainsi, si on observe une variation réelle faite d'une distribution spatiale de (x_i, y_i, d_i) ,

où (x_i, y_i) expriment les coordonnées géographiques d'un lieu i ,
 d_i mesure la densité au lieu i .

, si on considère le modèle de NEWLING (1969): $D_x = D_0 e^{(bx+cx^2)}$,

où D_x exprime la densité estimée à la distance x du centre du site urbain,

D_0 exprime la densité mesurée au centre du site ($x=0$),

x exprime la distance entre le centre et un lieu,

b et c sont les paramètres du modèle,

on estimera alors, par régression sur la variation réelle, b et c pour l'expression du modèle D_x . Une fois cette construction du modèle réalisée, on se propose d'étudier la variation modélisée.

On remarque donc la nécessité de l'introduction de traitements complémentaires d'application dans le logiciel décrit en II.2. afin de réaliser les calculs de modélisation.

II.3.2. Le projet-cadre de l'application géographique.

L'objectif est d'étudier la variation spatiale des densités urbaines et de permettre une modélisation d'une variation observée de densités.

L'étude comprend la saisie, visualisation et manipulation de la variation.

Globalement, la philosophie des objectifs définis dans le projet-cadre du logiciel général de manipulation de surfaces 3-D (paragraphe II.2.) reste valable, si ce n'est l'adaptation fonctionnelle due à la modélisation d'une variation réelle.

II.3.3. Spécifications globales complémentaires liées à l'application géographique.

On va ici énoncer les spécifications globales de l'application géographique en reprenant les quatre fonctions de la Figure I7, mais en insistant sur les particularités de l'application. Cela donne la découpe logique de la figure 21.

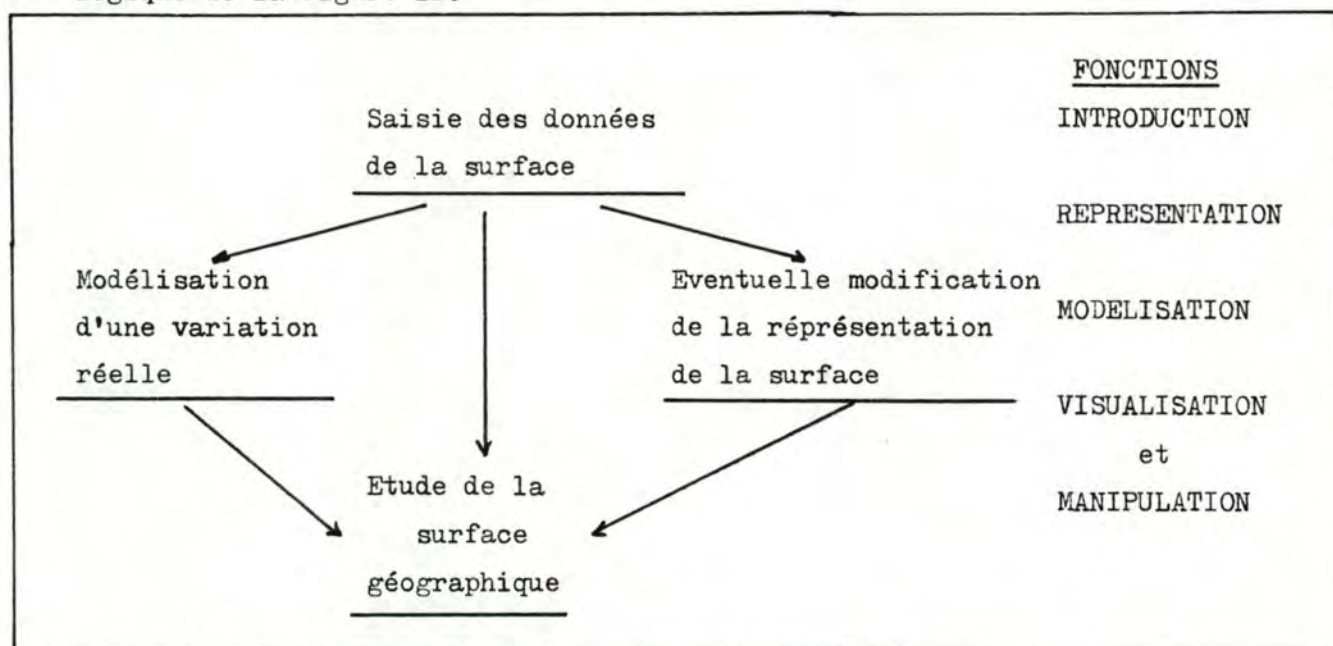


FIGURE 21. Les fonctions de base du logiciel appliqué à la géographie.

On y distingue:

- INTRODUCTION : (voir § II.3.3.1.1.)
- REPRESENTATION : au départ, la surface introduite sera une liste de points (densités) qui correspond à une surface de modélisation simple-valeur.
On implémentera donc la méthode de Brunet (1983) énoncée au paragraphe II.2.1.2.2.
- MODELISATION : cette modélisation revient en fait à introduire une nouvelle surface définie elle-même, par une équation (celle du modèle) (voir § II.3.3.1.3.).
- VISUALISATION : nous détaillerons cette fonction aux § II.3.3.1.2. et II.3.1.4. Nous avons séparé pour plus de clarté la visualisation d'une variation réelle, de celle d'une variation modélisée.
- MANIPULATION : cette fonction est semblable à celle du logiciel général (§ II.2.1.2.4.).

II.3.3.1. Par rapport aux fonctions particulières à l'application géographique.

II.3.3.1.1. Introduction de la surface.

La Variation réelle des densités peut être décrite sous 2 formes:

- $D(r)$ = densité de population à la distance r du centre,
- $P(r)$ = population totale comprise dans une zone circulaire de rayon r autour du centre.

D'un point de vue conceptuel, ces deux formes sont identiques:

$$P(r) = 2\pi \int_0^r D(r) dr.$$

Dans la littérature géographique, la plupart des modèles sont exprimés en fonction de $D(r)$. De plus, l'étude de modélisation est également réalisée à partir des $D(r)$ observés.

BUSSIÈRE et STOVALL (1971, P.22) proposent eux^{une} étude de modélisation à partir des $P(r)$, car elle est plus fiable.

- Dans ce contexte, nous faisons le choix de mener les deux "filières". L'utilisateur effectuera son choix dès le départ.

On devra donc introduire :

- Coordonnées du "lieu central" et des lieux de recensement.
- Densité en chaque centre de la zone de recensement. (1)
- Population de la zone de recensement (2)

Selon le choix de l'utilisateur, il fournira donc les données (1) et/ou (2) au progiciel. Son choix dictera la méthode de modélisation.

- A propos du centre, on doit pouvoir réaliser
 - une introduction interactive, mais aussi
 - un calcul selon la méthode de BUSSIÈRE ET STOVALL (1981, P.103).

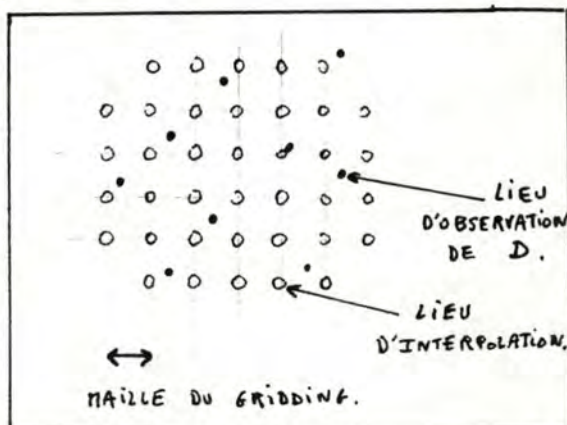
Dans le cas d'un axe central, seule l'introduction interactive serait prévue.

- Le calcul des distances (r) se fait par l'intermédiaire du système de coordonnées définit lors de l'introduction des "densités" et/ou "populations".
- L'utilisateur aura la possibilité de mémoriser les données introduits pour une région urbaine.

II.3.3.1.2. Visualisation de la variation réelle des densités.

a. A partir des valeurs de D observées.

- La surface 3-D de représentation doit présenter un aspect continu qui est obtenu par l'utilisation de la technique du "gridding". Ce "gridding" permet une représentation continue (et donc matricielle) de la surface. Il implique un lissage LOCAL afin de calculer les valeurs de densité aux noeuds du maillage.



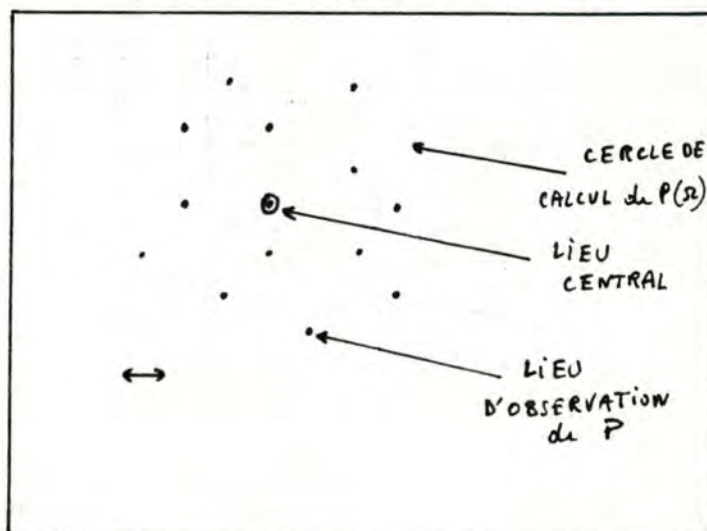
Ce calcul varie suivant la méthode d'interpolation.

- Trois méthodes d'interpolation locale peuvent être utilisées: JOURET (1976), BEGUIN (1979) et DELFINER, DELHOMME (1975). L'utilisateur pourrait choisir interactivement la taille de la maille ainsi que la méthode d'interpolation.
- La représentation type "bloc-diagramme" serait la plus efficace et devrait s'accompagner de :
 - . la représentation des axes de base du système de coordonnées,
 - . la rotation du "bloc-diagramme",
 - . la possibilité de coupes suivant des directions choisies,
 - . la représentation des limites de la région étudiée.
- A cette représentation 3-D, on peut ajouter le calcul d'isolignes de densités afin de préciser la visualisation. Ces isolignes seraient superposées au "bloc-diagramme" (Voir SPRUNT, 1973) ou feraient l'objet d'une visualisation dans le plan.

b. A partir des valeurs de P observées

- Cette représentation 3-D présente des différences avec la première.

Tout d'abord, elle nécessite un calcul préalable des $P(r)$ par la prise en compte de cercles concentriques (r croissant). Pour chacun de ces cercles, sont pris en compte les lieux d'observation de P inclus, afin de calculer le " $P(r)$ ". Il y a lissage préalable à la représentation 3-D.



Ensuite, pour chaque noeud du maillage, on affectera la valeur de $P(r)$ pour laquelle ce noeud est inclus dans le cercle de rayon r .

Ainsi, suivant le choix, de l'intervalle (Δr) dans le calcul des $P(r)$,
des "lieux centraux",

de la taille de la maille, une représentation

3-D des $P(r)$ sera possible. Ces choix seront interactifs.

- La représentation type "bloc-diagramme" serait la plus efficiente et devrait s'accompagner de :

- . la représentation des axes de base du système de coordonnées.
- . la représentation du "lieu central",
- . la rotation du "bloc-diagramme",
- . la possibilité de coupes suivant des directions choisies,
- . la représentation des limites de la région étudiée.

- Il serait plus parlant de représenter également la surface symétrique à celle des $P(r)$ par rapport au plan de base.

II.3.3.1.3. Modélisation d'une variation réelle.

L'objectif de cette modélisation est l'ajustement aux données de la variation réelle d'un modèle qui selon l'utilisateur lui semblera le meilleur. Une évaluation statistique de chaque ajustement l'aidera.

Pour cela il devra disposer du choix :

- .. des modèles théoriques
- .. des lieux centraux
- . de la méthode de modélisation.

En ce qui concerne les modèles, nous prenons le parti de trois modèles qui représentent les trois principales tendances de la littérature :

.. Modèle quadratique (NEWLING, 1969)

$$D_x = D_o e^{(bx + cx^2)}$$

. Modèle newtonien (AMSON, 1972)

$$D_x = D_o (1 + bx^2)$$

. Modèle directionnel (HAYNES et RUBE, 1973)

$$D_x = D_o e^{\frac{bx(\cos^2 \theta + a^2 \sin^2 \theta)}{a} + \frac{cx^2(\cos^2 \theta + a^2 \sin^2 \theta)}{a^2}}$$

où x = distance au "lieu central"

D_x = densité à la distance x

D_o = densité au "lieu central"

L'introduction des lieux centraux se ferait par la simple rentrée des coordonnées. Notons que la visualisation 3-D à partir des $P(r)$ a déjà requis telle introduction.

Enfin, en ce qui concerne le choix de la méthode de modélisation, nous laissons la liberté à l'utilisateur puisque l'introduction des données comprend à la fois les valeurs de P et de D . De nombreuses méthodes existent; on lira ZIELINSKI à ce propos. En fait, chacun propose sa méthode qui ne semble pas toujours justifiée.

Dans ce cas, le progiciel doit permettre l'introduction de ces différentes méthodes.

Toutefois, BUSSIÈRE et STOVALL (1981, P.23) montre le caractère indépendant et répétitif de la mesure des $P(r)$ sur l'ajustement d'un modèle.

Nous choisirons donc en premier lieu cette méthode pour réaliser la modélisation.

PROCEDURE.

A partir des données introduites de P ou de D , on réalisera un ajustement par la méthode des moindres carrés (OLS) après transformation pour obtenir un modèle simplement additif où la variable dépendante = Population cumulée ou densité, la variable indépendante = distance aux lieux centraux. On estimera ainsi les paramètres du modèle.

Chaque ajustement fera l'objet d'une évaluation statistique complète et claire pour l'utilisateur. En fonction de celle-ci, il décidera de choisir un autre modèle, d'autres ^{lieux} centraux, une autre méthode de modélisation.

II.3.3.1.4. Visualisation d'une variation modélisée.

Tous les modèles sont isotropes à partir des lieux centraux, on pourra donc facilement calculer la valeur de $P(r)$ estimée et/ou $D(r)$ estimée en chaque noeud de maillage de représentation.

En ce qui concerne la représentation 3-D à partir des $D(r)$ estimées et/ou $P(r)$ estimées, on peut faire les mêmes remarques concernant la représentation "bloc-diagramme", le calcul d'isolignes et l'inversion de la représentation.

On s'attachera également à visualiser dans le même système d'axes, les résidus d'une régression.

II.3.3.2. Remarques de spécification.

On devra envisager plusieurs comparaisons particulières.

- . Comparaison de surfaces réelles concernant deux structures de données
- . Comparaison de la surface-densité avec la surface-symétrique - population cumulée.
- . Comparaison des surfaces réelles et théoriques en densité et/ou population cumulée d'une même structure de données.

- Comparaison de surfaces théoriques concernant deux structures de données
Comparaison de la surface-résidu avec la surface réelle et/ou surface théorique d'une même structure de données.

Cette phase de comparaison consisterait en la superposition 3-D des 2 surfaces et en la soustraction de l'une par l'autre.

On pourrait envisager un tirage HARDCOPY des différentes surfaces,
une impression des résultats d'un ajustement,
une impression des données réelles modélisées de D et/ou P.

En termes de PORTABILITE, il serait bon que le logiciel soit indépendant du matériel graphique utilisé ^{et} portable pour produire les résultats sur micro-ordinateur.

En termes d'EXTENSIBILITE, il faut certainement prévoir:

- l'ajout possible de méthodes d'interpolation lors du gridding à partir des valeurs observées de D.
- l'ajout de modèles et notamment ceux faisant intervenir le temps.
- l'ajout possible d'une autre méthode de régression.

II.4. Cadre général de l'application chimique.

II.4.1. Les besoins liés à l'application chimique.

L'application chimique provient elle d'une demande des chimistes de la Faculté. Il s'agit d'étudier les surfaces moléculaires dont les points constitutifs sont issus du calcul réalisé par l'algorithme de Connolly (1982). Cet algorithme se nomme le "Molecular Surface Calculation (MSC)".

Le programme MSC nous fournit une surface de type POINTS,

MULTIPLE-VALUED définis par:

- les coordonnées des points de la surface CONTACT
- les coordonnées des points de la surface REENTRANT
- les coordonnées des vecteurs unitaires calculés sur la surface CONTACT.
- les coordonnées des vecteurs unitaires calculés sur la surface REENTRANT.

Qu'est-ce qu'une surface CONTACT et REENTRANT ?

En fait, Connolly calcule, à partir de la position des atomes de la molécule

et de la force de Van Der Waals correspondante à son type, deux types de points. Les points qui sont "influencés" par un seul atome et qui feront partie d'une surface convexe dite CONTACT, et ensuite les points "influencés" par plusieurs atomes (2 ou 3) et formant donc une surface concave dite REENTRANT.

Le schéma de la Figure 22 illustre ces deux types de points.

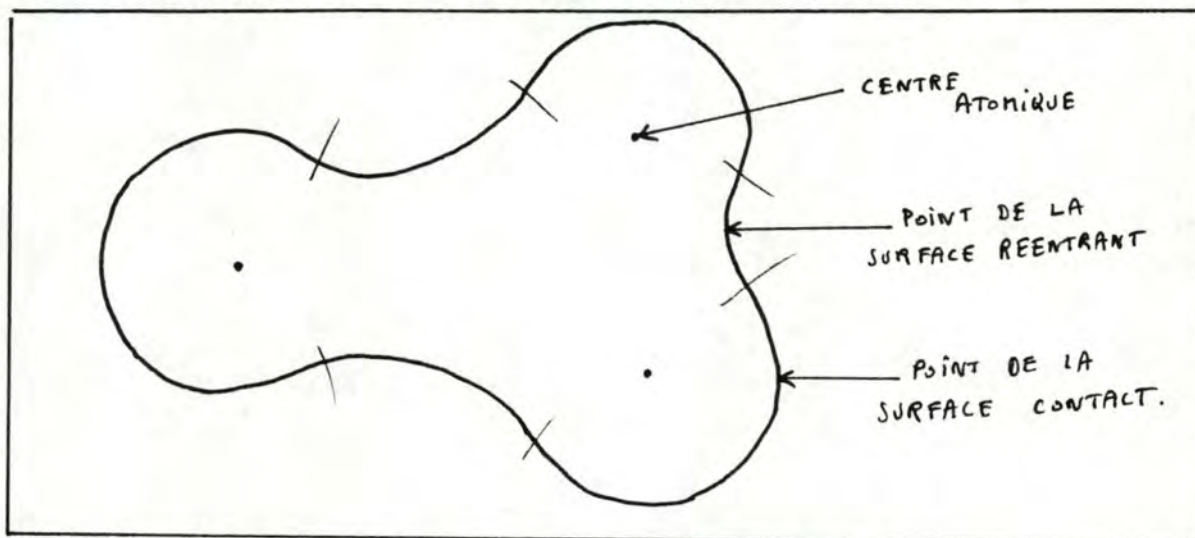


FIGURE 22 : Surface CONTACT - Surface REENTRANT.

La surface de Connolly est évidemment définie par les input du programme MSC :

- les coordonnées des atomes de la molécule
- le rayon de sondage pour le calcul des points.
- le type des atomes et leur force de Van Der Waals

correspondante.

II.4.2. Le projet-cadre de l'application chimique.

L'objectif est d'étudier les surfaces moléculaires issues du calcul MSC. L'étude comprend la saisie par le biais des outputs de MSC, la visualisation (de la surface moléculaire globale, des centres atomiques, des vecteurs unitaires indiquant le sens du sondage pour le calcul du point) et la manipulation de la surface.

Globalement, la philosophie des objectifs définis dans le projet-cadre du logiciel général de manipulation de surface 3-D (paragraphe 2.2) reste valable.

II.4.3. Spécifications globales complémentaires liées à l'application chimique.

Celles-ci sont très proches de celles du logiciel général. Soulignons simplement au niveau de :

- . l'INTRODUCTION : on doit permettre l'introduction des points de contrôle par le biais d'un fichier extérieur de points.
- . la REPRESENTATION : on devra utiliser la méthode des "parametric patches" vu le caractère multiple-valeur de la surface.
- . la VISUALISATION : on devra tenir compte de la visualisation de points non-éléments de la surface (centres atomiques et vecteurs unitaires).
- . la MANIPULATION : on réalisera les mêmes manipulations spécifiées en II.2.1.2.4.

II.5 En Conclusion.

Du projet cadre aux spécifications globales du logiciel général, nous avons élaboré la première étape de notre démarche de conception.

Les objectifs du projet-cadre sont ambitieux pour un mémoire, mais nous avons déjà restreint ceux-ci par le choix des types de surfaces 3-D à étudier.

D'autres restrictions seront énoncées au cours de notre démarche.

Au cours de ce chapitre, deux applications du logiciel général furent définies. Leurs objectifs s'insèrent dans ceux du logiciel général, leurs spécifications globales en sont très proches.

Nous voulions en effet que ce logiciel général soit ouvert au maximum d'applications spécifiques sans pour autant bouleverser les spécifications et objectifs de base. Mais l'application géographique notamment nécessitera quelques traitements complémentaires (calcul de modélisation).

Nous allons maintenant énoncer, au Chapitre III, les concepts ("Que faire?") à élaborer, pour répondre aux spécifications énoncées dans ce chapitre (" Quoi faire?").

CHAPITRE III.

L'Analyse Conceptuelle du logiciel général et ses applications

III,1 But et Méthodologie de l'analyse conceptuelle.

De façon rigoureuse, l'analyse fonctionnelle d'un projet informatique se décompose selon Bodart (1983) en deux étapes de base : l'étude d'opportunité et l'analyse conceptuelle (figure 23).

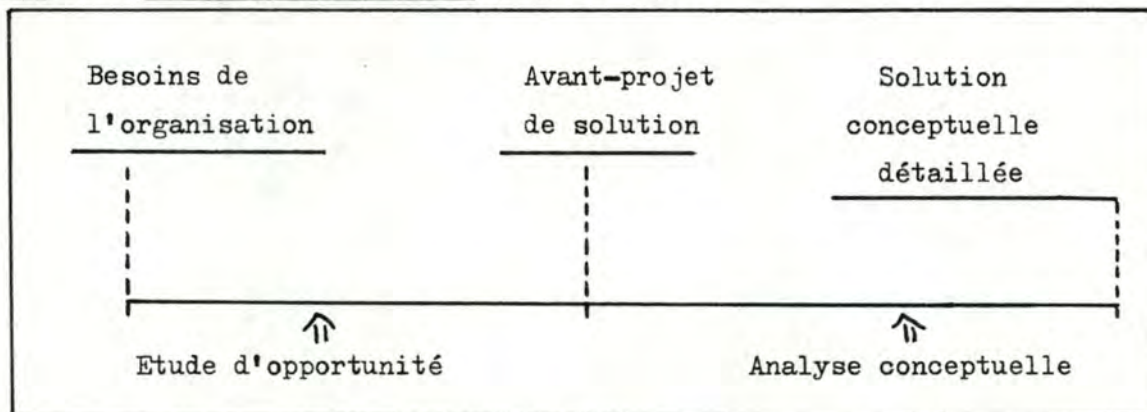


FIGURE 23. L'analyse fonctionnelle et ses résultats

D'après Bodart

Plus précisément l'étude d'opportunité comprend :

1. l'identification des besoins
2. l'élaboration du projet-cadre
3. la critique de l'existant
4. l'élaboration de solutions
5. le choix d'une solution

Par ailleurs l'analyse conceptuelle comprend elle :

1. l'élaboration par phase du projet, des sous-schémas conceptuels des données et des traitements.
2. la consolidation des sous-schémas en un schéma conceptuel global.

Au chapitre II, nous avons développé les étapes 1 et 2 de l'étude d'opportunité.

Pourquoi ne pas avoir effectué également les étapes 3, 4 et 5 ? Ceci s'explique facilement par l'absence de solution existante à critiquer sur laquelle des solutions pouvaient être élaborées, et parmi lesquelles un choix aurait été réalisé.

C'est pourquoi nous allons nous attacher à élaborer une "solution" de la façon la plus formelle possible mais en nous éloignant légèrement de la méthodologie des étapes 3,4 et 5 de l'étude d'opportunité.

Cette "solution" (Voir paragraphe III.2.) sera la base de l'analyse conceptuelle (Voir paragraphe III.3.) qui elle, suivra à la lettre la méthodologie proposée par Bodart (1983).

Rappelons que le but de l'analyse conceptuelle est d'aboutir à une solution conceptuelle sous forme de schéma.

Ce schéma conceptuel "est une description en termes de concepts et exprimée à l'aide d'un langage

- des informations

- des traitements associés et

- des conditions d'activation de ces dernières,

qu'il conviendra de représenter dans le futur système d'information pour satisfaire les besoins exprimés dans l'organisation ". (Bodart, 1983)

Concernant la méthodologie de l'analyse conceptuelle, rappelons que le sous-schéma conceptuel d'une phase est divisé en deux parties:

. l'une relative aux informations et leur structure,

. l'autre relative aux traitements et comprenant la décomposition de la phase en fonctions, l'expression de la dynamique entre ces fonctions et la spécification de la statique de chaque fonction. Ces trois démarches s'inspirent évidemment des modèles de la dynamique, de la statistique et de la structuration des traitements, eux aussi proposés par Bodart (1983).

Plus de détails méthodologiques sont évidemment disponibles dans l'ouvrage de Bodart (1983).

III.2. L'élaboration d'une "solution".

En vue de l'expression de cette "solution", nous allons dans un premier temps décomposer en phases (au sens du modèle de la structuration des traitements) le projet. Dans un second temps nous spécifierons la dynamique entre ces phases (selon le modèle de la dynamique des traitements). Enfin, en troisième lieu, nous détaillerons les objectifs et les contraintes de performances de chacune des phases identifiées.

III.2.1. La Décomposition en phases.

Selon le modèle de structuration des traitements, notre projet correspond à la réalisation d'un logiciel de manipulation interactive de surfaces 3-D et de ses applications géographiques et chimiques.

Ce projet ne comporte qu'une seule application (au sens bodart).

En effet, au sein de ce projet, on remarque un flux homogène d'informations concernant les surfaces étudiées. La division en plusieurs applications se justifierait donc difficilement et cela selon les critères de Bodart (1983).

Par contre, au sein de l'étude d'une surface, on peut, semble-t-il, distinguer plusieurs phases. Ceci dans la mesure où une étude de surfaces pourra comprendre plusieurs étapes (introduction, manipulation, ...) ayant des objectifs spécifiques et portant sur des structures de données différentes.

Il est donc opportun d'opérer une division de l'application en plusieurs phases. Ceci est encore confirmé par la définition d'une phase (traitement ayant une unité spatio-temporelle d'exécution) qui implique l'unité temporelle d'exécution (exécution sans interruption) pour celle-ci.

Et de fait, nous pouvons remarquer divers traitements ayant des unités temporelles d'exécution distinctes. Voyons cela.

Nous avons énoncé au chapitre II (figure 17), les fonctions de base du logiciel qui sont, rappelons-le, l'introduction d'une surface, son étude (visualisation et manipulation) et son éventuelle modification de représentation. Il semble évident que ces trois fonctions peuvent constituer autant de phases distinctes au vu de leur unité temporelle d'exécution : pour chacune de ces fonctions, on opérera sur une surface unique et, il semble essentiel de clôturer l'opération sans en interrompre le cours.

Ainsi par exemple, l'introduction d'une surface devra se faire en une fois, sous peine d'introduire une surface dépourvue de sens.

Il nous paraît également opportun de séparer l'étude d'une surface de celle de deux surfaces. Outre que l'on travaille sur des structures différentes et que l'on peut réaliser des opérations différentes, l'étude d'une ou deux surfaces sont distinctes dans le temps et ne peuvent se succéder directement sans passer par une phase de choix de surfaces.

A propos de choix de surfaces, les fonctions de bases citées ci-dessus devront être accompagnées d'un Coordinateur permettant, notamment, le choix de surfaces et d'opération sur celles-ci. Cette phase de choix s'identifie facilement par sa disconnection avec l'exécution des phases liées aux fonctions de base.

Avant et après l'étude d'une surface et la coordination de cette étude, une phase d'initialisation ainsi que de terminaison d'une session de travail nous semblent opportunes. La notion de session se distingue de l'étude proprement dite de surfaces. Une session pouvant ^{être} ouverte (et fermée) sans aucune étude de surfaces, il nous a donc semblé ^{utile} de créer deux phases, gérant cette session de travail.

Au sein de la version géographique, on a vu (paragraphe II.3.3.1.3) l'existence de la fonction particulière de modélisation d'une variation réelle de densités. Cette fonction est indépendante dans son exécution de l'étude d'une surface. Nous avons donc constitué une phase à part entière dite de traitements complémentaires.

Enfin, on peut énoncer la possibilité d'un état "parallèle" d'exécution du logiciel par le biais de la fonction d'aide à l'utilisateur (HELP) proposée dans les principes d'interfaçage (Voir Chapitre V).

A tout moment l'utilisateur pourra "interrompre" l'exécution de son travail et se renseigner sur l'état de celui-ci. On peut donc identifier cette phase comme particulière.

En résumé, la proposition de découpe en phases est la suivante:

- Phase n°1: Initialisation d'une session.
- Phase n°2A : Coordination - Version simple.
- Phase n°2B : Coordination - Version géographique.
- Phase n°2C : Coordination - Version chimique.
- Phase n°3 : Etude d'une surface.
- Phase n°4 : Etude de deux surfaces.
- Phase n°5 : Introduction, Consultation et Mise à jour d'une surface.
- Phase n°6 : Modification de représentation.
- Phase n°7 : Aide à l'utilisateur.
- Phase n°8 : Terminaison d'une session.
- Phase n°9B : Traitements complémentaires - Version géographique.

On remarque que cette proposition comporte 3 phases "Coordinateur".

En effet, les applications géographiques et chimiques induisent une coordination assez variable et où l'utilisateur préférera se trouver dans un contexte de coordination spécifique à l'application.

A ce stade il est intéressant d'avancer le concept de version.

Une version est associée à :

- . une liste d'utilisateurs de celle-ci,
- . une configuration de traitements nécessaires à la réalisation des objectifs de la version. (par exemple les phases 2B et 9B sont spécifiques à la version géographique).
- . une configuration de surfaces étudiables au sein de la version.

On distinguera donc trois versions : simple(A) géographique (B) et chimique (C).

Celles-ci seront liées à trois configurations de traitements.

On détaillera ces configurations en spécifiant au chapitre IV, les modules versionnels.

On distinguera enfin trois type de surfaces (simple, géographique, chimique) selon leurs liens à une version. Une surface ne pourra évidemment être liée qu'à une et une seule version. Une surface ne représente en effet qu'un seul phénomène.

III.2.2. La dynamique des Phases.

L'utilisation du modèle de la dynamique nous permet de décrire le schéma de la figure 24.

Remarquons que ce modèle n'est pas très explicite pour traduire la structure d'un "Case" au sens du Pascal. Nous avons donc adapté cette structure pour la figure 24 où l'on remarque l'enchaînement de conditions à partir du coordinateur.

Cette figure 24 nous amène à émettre quelques commentaires :

- . Le rôle du coordinateur de la phase 2 est illustré par l'ensemble des enchaînements conditionnels qui en sont issus vers l'ensemble des phases (hormis celle d'initialisation de la session)

- . On distingue les notions d'étude d'une surface et d'opération sur une surface. On entend par opération sur une surface, une action qui vise les éléments constitutifs (c'est-à-dire les points ou les équations de définition, et les renseignements attachés à la surface) de la surface afin de les créer, modifier, supprimer, visualiser et manipuler cette visualisation. On entend par étude d'une surface, une action qui vise seulement à visualiser les éléments constitutifs et à manipuler cette visualisation.

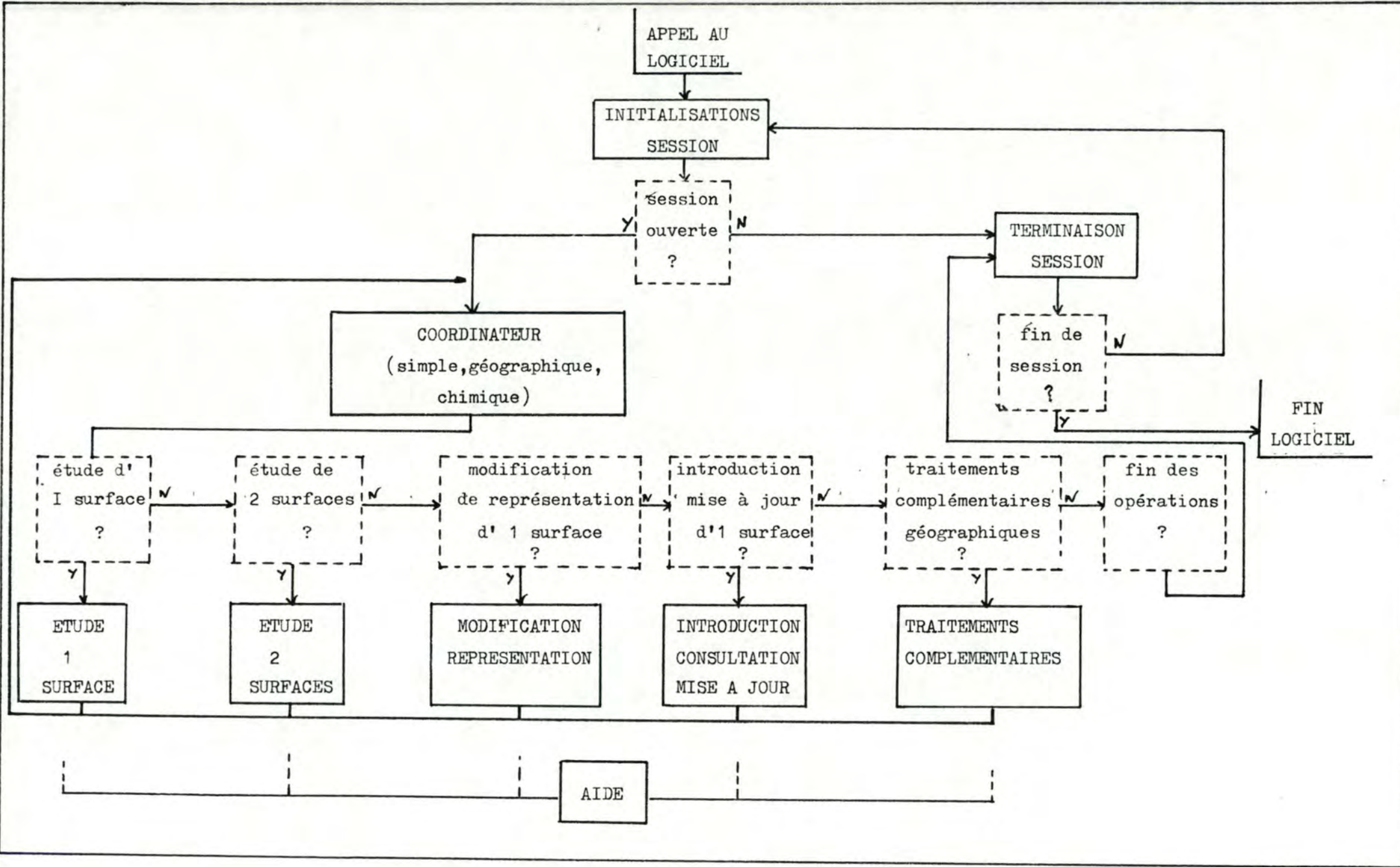


FIGURE 24: Dynamique des Phases.

Seuls donc les phases 3 et 4 seront des phases d'étude de surfaces.

. La phase d'aide est particulière dans le sens où elle peut être déclenchée par la simple présence d'un message d'appel (HELP). Elle entraîne alors l'interruption de la phase courante, quelque'elle soit, et l'activation de la phase d'aide. On peut schématiser la dynamique de la phase d'aide par la figure 25. On y remarque que l'appel à la phase d'aide provoque l'interruption de la phase i courante au profit de la phase d'aide (activation). La fin de l'appel d'aide provoque la désactivation de la phase d'aide au profit du redémarrage de la phase i.

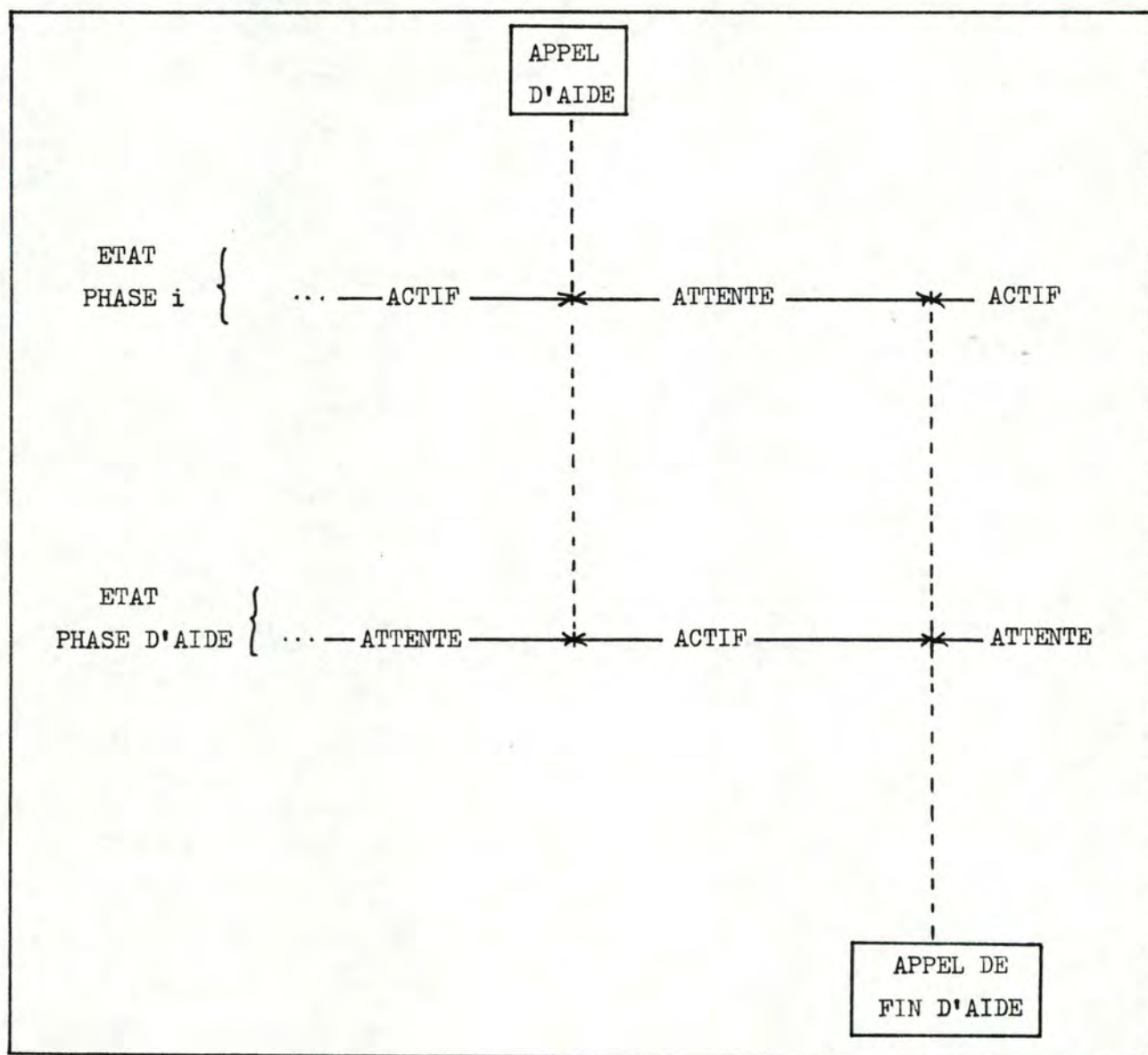


FIGURE 25 : La dynamique liée à la phase d'aide

III.2.3. Objectifs et Contraintes des phases.

Précisons maintenant les objectifs et les contraintes de performances de chacune des phases identifiées en III.2.1. On consultera en Annexe I par phase ces éléments de notre réflexion.

III.3. L'analyse conceptuelle.

III.3.1. Elaboration des sous-schémas conceptuels.

Sur base du projet de solution (paragraphe III.2.) et de sa découpe en phases (paragraphe III.2.1.), nous allons présenter en Annexe II, par phase :

- . le schéma conceptuel des informations établi par le biais du modèle E/A (Bodart, 1983)
- . le schéma conceptuel des traitements établi en quatre étapes:
 - 1) Présentation de la mise en oeuvre des objectifs de la phase, qui furent décrits en Annexe I.
 - 2) Présentation de la découpe en fonctions de la phase à partir des idées de mise en oeuvre de celle-ci.
 - 3) Présentation de la dynamique des fonctions
 - 4) Présentation des règles de traitement des fonctions.

Les étapes 2,3 et 4 utiliseront respectivement les modèles de structuration des traitements, de la dynamique des traitements et de la statique des traitements.

III.3.2. Consolidation des sous-schémas conceptuels.

III.3.2.1 Au niveau des données.

On trouvera à la figure 26 le schéma global des données. Ce schéma résulte de la mise en commun des sous-schémas tout en évitant les erreurs d'intégration.

III.3.2.2 Au niveau des traitements.

On se rapportera aux schémas partiels décrits en annexe II et s'intégrant dans le schéma de la figure 24.

III.4 Conclusion.

Hormis la phase de consolidation, nous avons présenté ici l'ensemble des éléments de base d'une analyse fonctionnelle. Ses résultats permettront d'aborder de façon correcte les problèmes d'architecture.

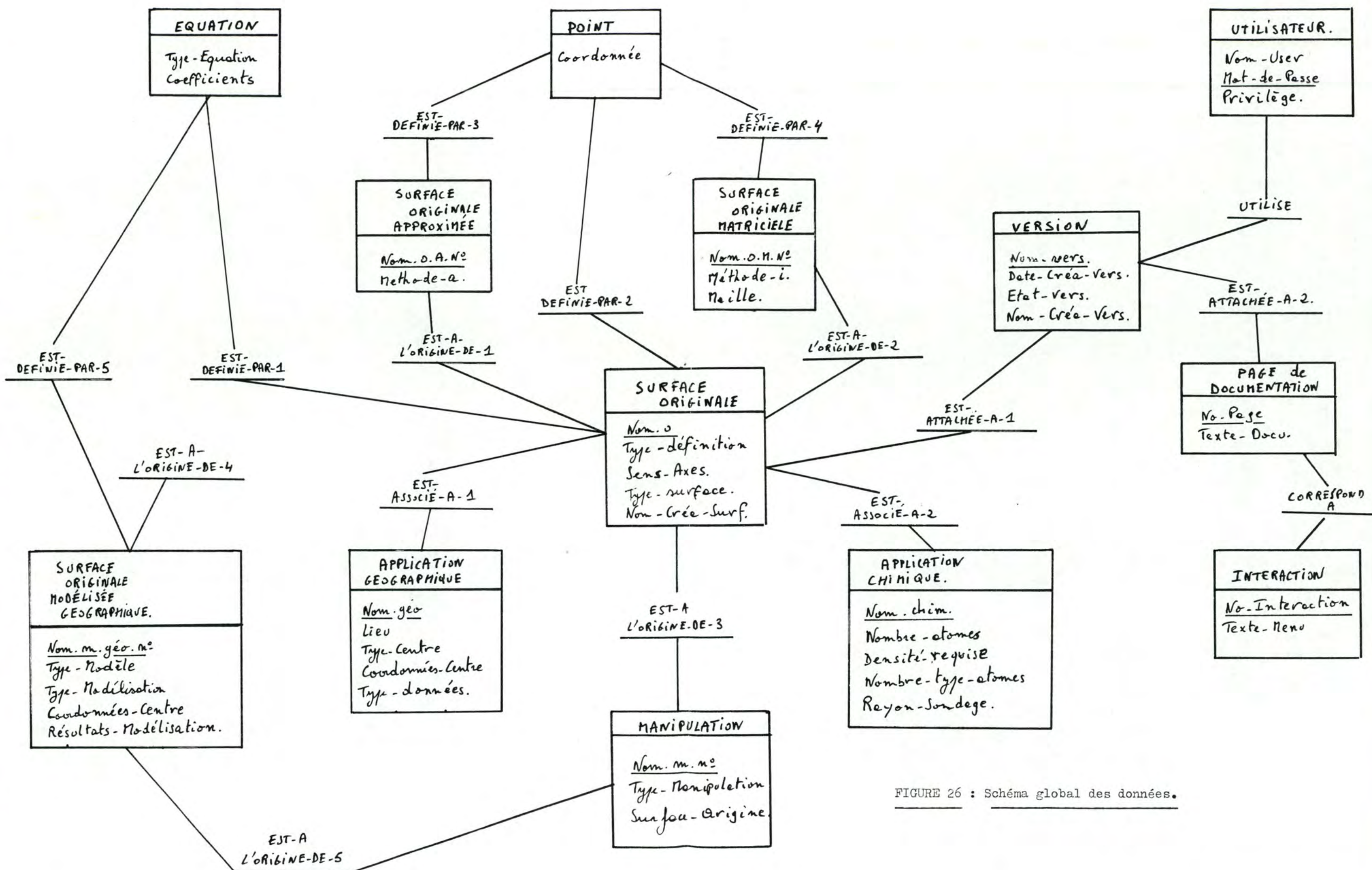


FIGURE 26 : Schéma global des données.

CHAPITRE IV.

UNE ARCHITECTURE LOGIQUE GLOBALE DU LOGICIEL

IV.1. Introduction.

Comme nous l'avons exprimé au paragraphe I.2.1., structurer notre système de façon hiérarchique et modulaire selon la méthode de VAN LAMSWEEERDE (1983), nous permet d'obtenir un logiciel de qualité. Cette structuration hiérarchique et modulaire conduit alors à élaborer une architecture logique liée au logiciel. En effet, Van Lamsweerde (1983) précise qu'établir une architecture logique revient à structurer le système (logiciel) en un ensemble de composants, ou encore modules

- . qui devront comprendre les données et les traitements de l'analyse conceptuelle, les entrées/sorties, les accès à la BD, ...
- . qui devront comporter entre eux des relations bien définies (UTILISE, IMPORT-EXPORT, ...)
- . qui auront les propriétés requises de validité, robustesse, ...

Mais comment hiérarchiser et modulariser notre logiciel ?

La hiérarchisation consiste à définir, à partir des composants fonctionnels (car issus de l'analyse fonctionnelle), une hiérarchie de composants (dits "modules") reliés par une relation hiérarchisante, ici la relation UTILISE. On a vu l'intérêt de cette relation au paragraphe I.2.1.

Cette hiérarchie garantit notamment une meilleure séparation des problèmes : les modules des niveaux supérieurs pourront en effet utiliser les primitives offertes par les composants inférieurs, et la conception des premiers sera exempte des problèmes résolus par les seconds.

On le voit, et de façon prévisible, la hiérarchie implique la création de niveaux de composants. A chaque niveau sera affecté des composants suivant leur utilité. La répartition de base des composants est celle de la figure 11. Mais nous devons tenir compte des particularités de notre LGI et notamment des modifications proposées au paragraphe L3.2. Au paragraphe IV.2., nous détaillerons nos choix et leurs justificatifs.

Ensuite vient la modularisation. Celle-ci consiste, par niveau hiérarchique, à identifier les composants ou modules, nécessaires selon l'analyse fonctionnelle (Chapitre III). Cette étape est donc cruciale.

C'est au paragraphe IV.3 que nous présenterons nos choix et leurs justificatifs. Notons de suite que les principes d'interfaçage (Chapitre V) influenceront également la modularisation.

Enfin, nous devons spécifier les objectifs de chacun de ces modules, c'est-à-dire, les spécifications abstraites (Voir paragraphe IV.4.) connues de tous les concepteurs. Nous pourrions alors aborder cette conception au chapitre VI.

Le résultat final de cette structuration du logiciel consistera donc en une sorte de graphe logique d'architecture où les noeuds seront les modules spécifiés et les axes, les relations entre modules.

Au paragraphe IV.5 nous décrirons ce graphe.

Ce chapitre sera clôturé par la présentation de choix préalables à la conception, nécessaires au vu des contraintes de temps et ^{de} complexité qu'implique la réalisation globale du logiciel (Paragraphe IV.6.)

IV.2. La Hierarchisation du logiciel.

La hierarchisation du logiciel, suivant les principes de Van Lamsweerde (1983) et présentant la répartition des composants de la figure 11, doit tenir compte des particularités des LGI. Rappelons les :

- . l'importance de l'interface Homme-Machine,
- . la diversité des moyens techniques d'E/S et la modularisation des fonctions de LGI ainsi nécessaire.

Ceci nous amène à proposer la hierarchisation suivante :

- . Au niveau 6 : Composants fonctionnels issus de l'analyse conceptuelle et plus particulièrement de son sous-schéma conceptuel des traitements.
- . Au niveau 5 : Composants de base notamment liés aux transformations nécessaires avant visualisation d'une surface (vu et caché et transformations reprises au paragraphe II.2.1.2.3.), à l'historique des opérations réalisées par l'utilisateur, à l'historique des manipulations réalisées sur une surface, à la réalisation de séquences fréquentes de dialogue et d'interfaçage.

Ces composants pourront être, lors de l'utilisation du logiciel, à tout moment utilisés, d'où leur qualificatif de "base".

Par exemple, il nous paraît important de créer un composant de calcul des coordonnées-écran d'une surface afin de la visualiser. Ce composant, à partir de la liste d'affichage actuelle, d'un paramètre de modification à appliquer, devra recalculer la liste d'affichage liée à la surface ainsi modifiée. (voir module n°53).

- . Au niveau 4 - Composants d'accès à la BD permanente contenant les données et renseignements généraux de chaque surface présente dans le système. Cette BD sera construite à partir du schéma conceptuel des données de l'analyse conceptuelle.

- Composants de gestion des zones de l'écran de visualisation et des moyens techniques d'entrée. Ces composants seront vus par le biais des primitives de haut-niveau gérant les moyens techniques d'E/S, indépendamment de l'origine de construction de ces moyens techniques. Ainsi par exemple, pour la souris, on créera un module visible par ses primitives de positionnement du curseur, d'activation du curseur (visible), de mise en état "activable" (on peut introduire une information à partir de la souris) ou "non activable", de renvoi de l'information demandée.

- . Au niveau 3 : Composant de coordination du système et de son exécution. Nous avons choisi pour éviter une part trop grande de complexité de ne réaliser à ce niveau, que la coordination inter-phases et non la coordination entre toutes les fonctions (au sens de Bodart). Les composants fonctionnels du niveau 6 réaliseront la coordination inter-fonctions.

IV.3. La Modularisation du logiciel.

Le but de ce paragraphe est de préciser, par niveau de la hiérarchie,

- . les modules qui y seront attachés(en accord avec les notions du paragraphe IV.2.)
- . la justification de la création de ces différents modules.

Le nom du module éclairera le lecteur sur ses objectifs. On consultera toutefois le paragraphe IV.4 et l'annexe II afin de prendre connaissance des spécifications abstraites claires et exhaustives de chaque module.

Auparavant, précisons les notions importantes de modules et de modules versionnels.

Un module est un composant d'une architecture dont les attributs (spécifications, algorithmes, interfaces, performances) sont définissables de manière simple et précise. De plus, un module offre les caractéristiques suivantes :

- . une forte capacité à cacher de l'information,
- . une forte cohésion interne,
- . un faible degré de couplage, c'est-à-dire, du nombre de connexions avec d'autres modules.

Un module versionnel est ^{un} module dont le comportement et les attributs varient quelque peu suivant la version du logiciel (on le notera par "MV2" dans les tableaux 1 à 4) ou un module qui n'est nécessaire que dans le cadre d'une version précise (on le notera par "MV1" dans les tableaux 1 à 4).

IV.3.1. Au niveau 6.

Le tableau 1 montre la correspondance entre les numéros des traitements de l'analyse conceptuelle (fonctions au sens Bodart) et, les noms des modules à créer.

Pour plus de clarté, nous reprenons au tableau 2 la correspondance entre

les numéros des traitements et,
les noms de ceux-ci.

Examinons les correspondances du tableau 1

De façon générale, chaque fonction de l'analyse conceptuelle a induit la création d'un module à ce niveau. Mais l'on pourra avoir des regroupements de plusieurs fonctions au sein d'un même module, et ceci dans le cas de fonctions similaires et de phases différentes, de fonctions liées à un même résultat, ou encore liées à un même concept. La découpe d'une fonction en plusieurs modules pourra également avoir lieu, si cette fonction recouvre des traitements distincts (en vertu de la forte cohésion interne d'un module requise).

Une exception à ces règles est la fonction d'aide : utilisables par toutes les fonctions pour leur dialogue, elle sera affectée au niveau 5.

Ainsi on remarque les regroupements dans les cas suivants:

- les modules n°63,65 et 66.Ceci a pour origine la meilleure cohérence interne induite par ces regroupements.En effet,les fonctions constitutives de ces modules sont liées au concept unique d'entrée et de validation des choix de base de l'utilisateur,avant opération sur une surface.
- les modules n°610,611 et 64.Ici on regroupe des fonctions identiques de phases distinctes.Il est intéressant de noter la particularité du module 614 : il regroupe des fonctions de phases différentes mais issues des trois versions du logiciel.L'analyse conceptuelle et son point de vue utilisateur, a cru bon séparer en trois phases la coordination des trois versions.Mais on peut,dans un point de vue algorithmique, identifier des éléments communs,quelque soit la version (ici le chargement des données de la surface originale).

De même,on a séparé la fonction d'enregistrement de données en deux modules : les modules n°616 et 617.Cette scission repose sur les différences d'une part,de données à introduire(Point/Equation) et,d'autre part de méthode d'introduction(variée/unique-ment au clavier)

IV.3.2. Au niveau 5.

Le tableau 3 reprend les modules identifiés au sein du niveau 5.On l'a vu d'après la nomenclature du paragraphe IV.2.,ces modules doivent correspondre à des composants de base assurant des fonctionnalités communes à tous les modules du niveau 6.C'est donc à la suite d'une analyse approfondie des fonctions de l'analyse conceptuelle et des modules du niveau 6, que nous avons élaboré la liste du tableau 3.Mais comme on le verra pour certains modules,les principes d'interfaçage susciteront aussi leur création.

Notons trois types de modules au niveau 5 :

. Les modules n°52 et 53 réalisent des opérations primordiales qui, à partir de données de base de la surface et de son état actuel de manipulation, produisent la liste d'affichage de la surface après les transformations nécessaires. Nous avons séparé les transformations nécessaires à la visualisation nommées "Vu et caché" des autres.

En effet, la visualisation avec élimination des parties cachées (Vu et caché) nécessite un calcul "batch" (vu la longueur des calculs) et ne pourra se faire qu'en fin de session; au contraire des autres transformations qui elles seront réalisées en "temps réel".

Pourquoi n'avoir pas séparé ces dernières transformations en des modules distincts? Nous avons jugé celles-ci suffisamment inter-reliées dans leur nature. De plus on cache mieux l'information en créant ce module qui est simplement la "boîte à visualiser" d'une surface et ceci quelque soit le type de transformations implémentées. Par contre, en termes de degré de couplage, celui-ci est plus important: c'est un compromis qu'il nous fallait trancher.

. Les modules n°54 à 57 permettent eux la mise en oeuvre de commandes disponibles à l'utilisateur, dans le cadre d'une interface agréable :

- la commande de retour en arrière (avant modifications) pour une surface (module n°54)
- la commande de retour à la situation précédant la dernière commande (module n°55.)
- la commande d'aide à l'utilisateur (module n°56)
- la commande de mémorisation d'une surface après une phase de modification (module n°57.)

Remarquons que la fonction d'aide, au sens Bodart, parce que de base, fut attachée au niveau 5 et non 6.

. Le module n°51 est issu des principes d'interfaçage (Voir chapitre V). Ceux-ci nous montrent l'existence de séquences d'"écrans" habituelles, dont seuls certains paramètres changent (le contenu des menus, par exemple).

Il nous^a donc semblé utile de créer un module de gestion de ces séquences.

On le voit, les principes d'interfaçage qui seront énoncés au chapitre V, influencent la modularisation. Dans ce chapitre V, nous préciseront encore l'utilité de ces modules nécessaires à l'interface et qui se situent au niveau 5, mais aussi, on va le voir, au niveau 4.

IV.3.3. Au niveau 4.

Le tableau 4 liste les modules affectés au niveau 4. La hiérarchisation (paragraphe IV.2) affectait à ce niveau les composants d'accès à la BD et les composants de gestion des moyens techniques d'E/S.

Examinons les modules de ces deux catégories et justifions leur présence.

IV.3.3.1. Les Modules d'accès à la BD.

De l'analyse du schéma conceptuel des données (figure 26) on peut identifier cinq groupes de données permanentes au sein du projet. Chacun de ces groupes induira un module réalisant les accès en lecture, écriture et suppression des données associées au groupe.

Ces cinq groupes de données sont :

- . Données liées aux surfaces dites "ORIGINALES", surfaces introduites par un utilisateur, avec ou sans modification de représentation (module n°41).
- . Données liées aux "MANIPULATIONS", manipulations issues d'une surface originale après manipulation (module n°42.).
- . Données liées aux surfaces dites "ORIGINALES MODELISEES GEOGRAPHIQUES", surfaces issues, dans la version géographique, d'une surface originale ayant subi une modélisation (module n°43).
- . Données liées à la gestion de ma commande HELP et de la documentation (module n°417),
- . Données liées à la gestion du logiciel (module n°418).

Justifions cette découpe en cinq groupes.

Les trois premiers groupes correspondent aux données liées aux trois types de base de surfaces. Le critère principal d'identification de ces trois types de surfaces est la forte cohésion interne en termes de types de données et de leur structure. Les trois principes suivants prouvent la cohérence interne, forte autour des trois types de surface (Figure 27) :

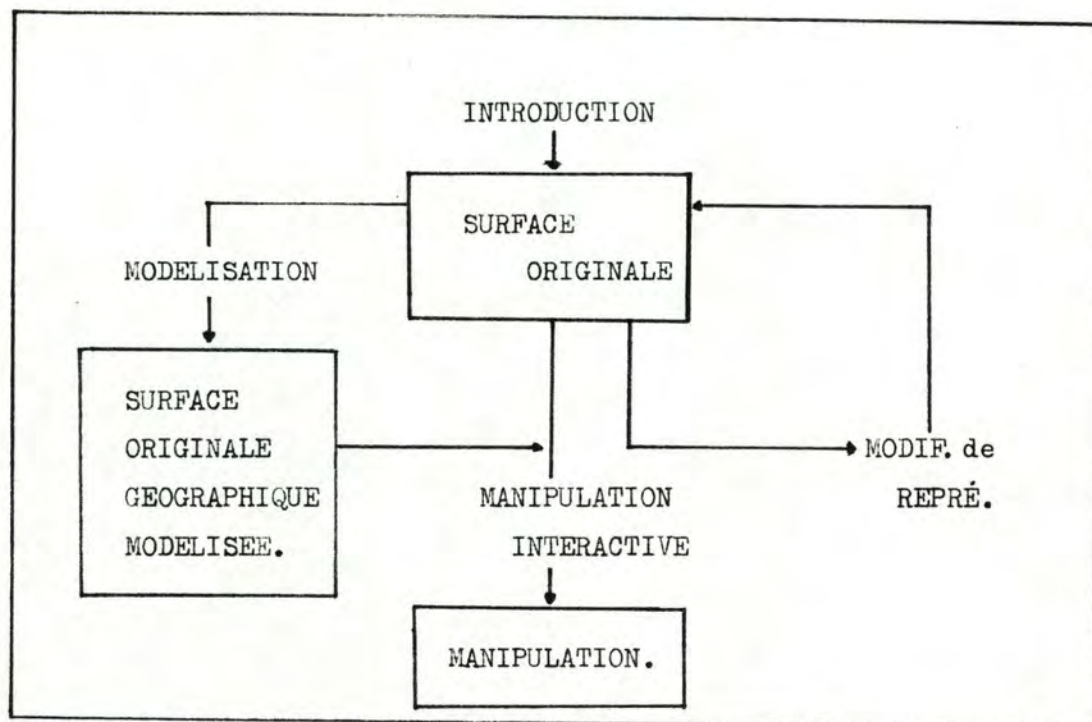


FIGURE 27 : Les types de surfaces et leur cycle de vie.

. La modification de représentation ne change pas fondamentalement la nature et la forme du phénomène 3-D exprimé par une surface originale. Les surfaces originales pourront même après modification de représentation, être considérées comme originales.

. La manipulation d'une surface donne naissance à un phénomène 3-D de même nature mais de forme différente. Ce type de surface s'individualise donc et nécessite un module distinct.

. La modélisation d'une surface originale géographique donne lieu à la création d'une surface de forme différente. Ce type de surface s'individualise donc également et nécessite aussi un module.

Remarquons que cette cohérence de structure de données nous permettra d'associer un type abstrait à chaque module (Voir § IV.4.1)

Les deux derniers modules (n°4I7 et 4I8) expriment l'accès aux données permanentes nécessaires à

- . la commande d'aide (HELP) et à sa documentation accessible.
- . la gestion du logiciel (listes d'utilisateurs, de mots de passe, renseignements concernant les versions du logiciel disponibles).

Les différences de structure de données et d'objectifs visés par chaque structure nécessitent la mise en oeuvre séparée des modules d'accès.

4.3.3.2. Les modules de gestion des moyens techniques d'E/S.

Parmi les composants de gestion des E/S, on peut logiquement déceler ceux qui s'occupent des entrées d'une part, des sorties d'autre part!

L'entrée d'informations nécessaires au logiciel sera gérée par les modules :

- n°44, pour la saisie et la validation syntaxique des introductions au clavier,
- n°412, pour la gestion des introductions par le biais de la souris,
- n°414 pour la gestion des introductions par le biais de la boite à fonctions,
- n°415, pour la gestion des introductions par le biais de la table de digitalisation.

La diversité des primitives offertes par chaque type de moyens techniques nécessite la distinction d'un module par moyen.

Enfin un module de validation sémantique de ces introductions (n°45) sera créé. Nous avons jugé que la cohérence de traitement (validation) et la capacité à cacher de l'information étaient suffisantes. Cela malgré la diversité des structures de données à valider.

La sortie d'information elle, sera gérée par les modules
n°46 à 411 et 413, pour les différentes zones de
visualisation,

n°416, pour la gestion d'un terminal standard.

On trouve dans cette modularisation, un deuxième exemple de l'importance de l'interface.

Ainsi chaque zone de l'écran de visualisation identifiée dans les principes d'interfaçage (Voir chapitre V) sera à la base d'un module de gestion. En effet, les différentes zones peuvent supporter des opérations variées et spécifiques. Le critère de cohésion interne nous pousse donc à implémenter ces différents modules.

4.3.4. Aux niveaux 3 et 2.

C'est au niveau 3 que nous avons placé un module de coordination (n°3I) comme le préconise Van lansweerde (figure 11).

Au niveau 2, vu l'importance des LGDB dans les LGI, nous implémentons un module dont les primitives facilitent la gestion des moyens techniques d'E/S. Les primitives offertes seront évidemment de bas-niveau et seront utilisées principalement par les modules des niveaux 4 et 5.

Ce LGDB devra au moins comporter les primitives proposées par Newman et Sproull (Voir paragraphe I.2.2.1.2.). La richesse des primitives du LGDB ne pourra qu'influencer la facilité de programmation ou éventuellement celle de conception des algorithmes des modules de l'architecture logique.

Nous exprimerons au chapitre VI, le choix de LGDB envisagé dans le cadre physique de notre logiciel. Pourquoi dans ce cadre? Car en l'absence de standardisation, les LGDB sont encore dépendants du matériel et donc leur choix doit s'y relier.

Rappelons ici que notre démarche de conception et son architecture logique permettront une forte indépendance du logiciel vis-à-vis du LGDB. Seul le codage (et éventuellement certains algorithmes) serait affecté par un changement de LGDB ou de matériel. La portabilité semble donc ici favorisée. La technique de l'architecture logique est notamment faite pour cela!

IV.4. Les spécifications abstraites des modules.

IV.4.1. Rappels méthodologiques.

Pour être certain que le module fait ce qu'il doit faire, le concepteur a intérêt à préciser (spécifier) ce qu'il veut que ce module fasse. Ces spécifications devront être connues de tous les concepteurs, si il y a une répartition de travail de conception entre plusieurs personnes.

De plus les spécifications formelles seront préférées lors du choix d'une méthode de spécification. Car elles seront une base possible à des tests élaborés (Voir Chapitre VII).

Nous avons choisi comme méthode de spécification la technique de spécification par assertions, encore appelées méthode par "pré et post-conditions". Le lecteur consultera Van lansweerde (1983) pour connaître les principes de cette méthode.

Un des attributs d'un module est son interface (Voir §IV.3.). On distingue deux types de modules en regard du type d'interface:

- . les modules de traitements visibles par leurs arguments et les résultats produits.
- . les modules de données visibles par leur opération sur une structure de données.

Aux modules de traitements sera appliquée la méthode de spécification par assertions comme prévu.

Aux modules de données sera, avant spécification, affecté un type abstrait. Un type abstrait est une structure de donnée associée à un concept de base du problème et entièrement caractérisée, et visible par

- les propriétés inhérentes à cette structure,
- les opérations réalisables sur cette structure,
- les spécifications de ces opérations.

La spécification d'un module de données reviendra donc à spécifier un type abstrait. Pour ce faire, on spécifiera la structure de données (ses propriétés) et ensuite les opérations sur cette structure, par la même méthode de spécification par assertions. On trouvera principalement les modules de données au niveau 4 de notre hiérarchie. Au tableau 4 nous avons noté par "MD" les modules de données.

IV.4.2. Les spécifications abstraites.

On trouvera en annexe III, les spécifications abstraites des modules identifiés dans ce chapitre.

Le module LGDB lui sera décrit au Chapitre VI. Remarquons que les spécifications, des modules issus des principes d'interfaçage, sont évidemment dépendantes de ceux-ci. Certaines spécifications faites au Chap. IV sont donc dépendantes de considérations faites au Chapitre V.

IV.5. Un graphe d'architecture logique de notre système.

Ce paragraphe a pour but de compléter le graphe de l'architecture logique et plus particulièrement les "arcs" du graphe, c'est-à-dire, les relations entre les "noeuds" (modules).

La relation de base selon le principe de hiérarchisation, est la relation UTILISE.

Mais d'autres relations seront nécessaires :

- . la relation d'ACTIVATION : un module déclenchera l'exécution d'un module à la fin de son exécution propre.
- . la relation d'IMPORT-EXPORT : un module exportera des arguments dans le but d'une exécution d'un autre module qui lui importera ses arguments.

Le graphe et ses relations multiples sont, pour raison de clarté, signifiée par le biais du tableau 5.

IV.6. Les choix préalables à la poursuite de notre travail.

Nous allons ici exprimer la deuxième restriction de notre mise en oeuvre du logiciel. Celle-ci est fondamentale. Elle concerne à la fois, la poursuite des spécifications de l'interface (Chapitre V) et aussi la conception, le codage et la conduite des tests (Chapitre VI et VII).

Certaines des fonctions de l'analyse conceptuelle ne seront pas réalisées, ainsi donc que les modules correspondants.

Ceci dans le but à la fois de permettre l'implantation d'un "core" de logiciel et de réduire la complexité de certains problèmes en les simplifiant.

Notre temps de travail est en effet limité.

Nous aurions pu développer une autre stratégie pour la réalisation du mémoire : Réaliser de façon complète et progressive les étapes de notre démarche. Dans ce cas, notre planning ne nous donnait aucune chance de pouvoir concrétiser, par un produit exécutable, notre travail.

La stratégie de restriction elle, nous permettra peut-être d'atteindre un objectif important de notre travail : tester si la démarche est applicable dans toutes ses étapes, mêmes celles de codage et mise en machine. L'étape qui visait à établir une architecture logique globale du système est donc la dernière qui sera réalisée complètement. Cela permettra une éventuelle reprise du travail dans un but d'achèvement. En effet, à partir des spécifications abstraites, un concepteur pourra implémenter les modules manquants assez facilement.

Après l'étape de ce chapitre, nous allons donc restreindre nos objectifs. Pour ce faire, nous proposons de ne pas implémenter :

- . la possibilité de mémorisation d'une surface manipulée,
- . la possibilité de modification des caractéristiques de visualisation d'une surface,
- . la transformation vu et caché,
- . l'étude de deux surfaces (comparaison spatiale et statistique),
- . la possibilité de modification de représentation d'une surface,
- . dans la version géographique, les traitements complémentaires à la modélisation.

- . la possibilité d'introduire une surface non-mathématique multiples-valeur,
- . les fonctions d'aide (HELP) et de retour à la situation avant la dernière commande (IGNORE),
- . la mise à jour des données et renseignements introduits et validés,
- . certains principes d'interfaçage (Voir Chapitre V).

Ces restrictions, importantes sans doute, furent choisies afin de permettre malgré tout :

- . d'introduire une surface définie par des points ou une équation, en simple-valeur,
- . de la visualiser,
- . de la manipuler interactivement.

Les objectifs de base, tels qu'ils ont été définis au paragraphe II.1.2, seront donc atteints, au moins partiellement.

IV.7. CONCLUSION.

En accord avec les principes de notre démarche de conception, nous avons élaboré dans ce chapitre une architecture globale.

Cette architecture globale du logiciel assure la mise en oeuvre des fonctions de l'analyse conceptuelle, mais aussi les qualités du produit et certains principes d'interfaçage. Cela est spécifique à notre démarche et aux LGI.

Notons que les quelques cinquante modules identifiés pourront faire l'objet dans le cadre de leur conception algorithmique, d'une subdivision si leur complexité s'avérait trop grande.

CHAPITRE V

SPECIFICATION DE L'INTERFACE HOMME-MACHINE

V.1 But et Rappels méthodologiques de la spécification d'un interface.

Au chapitre I, nous avons présenté la spécification de l'interface comme un acte essentiel d'une démarche de conception d'un LGI.

Nous allons nous appliquer à ce design dans ce chapitre.

La spécification d'un interface pose deux problèmes fondamentaux :

- . Quelle méthode utiliser pour spécifier l'interface?
- . A quel stade intégrer cette spécification au sein d'une démarche classique de conception d'un logiciel?

Les considérations des paragraphes I.2.2.2 et I.3.2 ont jeté respectivement les bases d'une réponse à ces deux questions.

La méthode de spécification à employer correspond à la mise en oeuvre d'une combinaison des méthodes de Green (1981) et de Newman et Sproull (1979).

Cette combinaison nous amène à préciser par unité de spécification :

1) TASK MODEL (USER MODEL) : Spécification des objets manipulables par l'utilisateur et des actions (tâches) réalisables sur ces objets. Certaines tâches seront non-interactives : l'utilisateur n'interviendra pas dans leur réalisation.

Ces tâches seront évidemment moins intéressantes à étudier dans le cadre d'un interface.

2) CONTROL MODEL (COMMAND LANGUAGE) : Spécification du langage de commande pour réaliser les tâches interactives du TASK MODEL.

Dans ce CONTROL MODEL, on spécifiera également des objets de contrôle (menus, curseur, ...) qui permettront la mise en oeuvre des commandes.

3) INFORMATION DISPLAY : Spécification du "look" de l'exécution des différentes tâches du TASK MODEL.

4) FEEDBACK : Spécification des feedbacks et des conditions d'erreurs pouvant amener des messages d'erreur (voir paragraphe V.2.2.4)

Nous avons déjà présenté ces étapes au paragraphe I.2.2.2.

Préciser le stade d'intégration de la spécification de l'interface revient à définir les unités de spécification.

Ces unités seront-elles des fonctions de l'analyse fonctionnelle, des modules d'un niveau de l'architecture logique, ou encore l'ensemble des modules de cette architecture?

Green prône comme unité de spécification les modules de l'architecture logique.

Par définition, ces modules sont en effet les unités de spécification. On peut donc logiquement y adjoindre les spécifications de l'interface. Mais, on doit tout d'abord remarquer que l'interface homme-machine a pour lieu de mise en oeuvre les modules fonctionnels du niveau 6. C'est en effet à ce niveau que se gère l'interaction homme-machine. Remarquons que ce fait n'empêche pas que la gestion de l'interaction nécessite l'utilisation des modules de niveaux inférieurs.

On doit remarquer également que l'architecture logique, dans sa modularisation et sa hiérarchisation, avant la spécification des modules, est influencée par la mise en oeuvre de l'interface (voir chapitre IV).

Ainsi, les principes retenus dans la conception de l'interface entraînent la création de modules de gestion des moyens techniques d'E/S, ou encore de gestion des zones des écrans d'affichage. Ceci pousse donc à définir les principes d'interfaçage avant la hiérarchisation et la modularisation.

En résumé, on peut proposer la méthode originale d'intégration de la spécification de l'interface, dans une démarche classique, suivante :

1) Définir les principes d'interfaçage (tels qu'ils le seront au paragraphe V.2).

Ces principes regroupent les idées directrices de l'interface homme-machine (type et moyens de dialogue).

Ces principes d'interfaçage sont semblables au projet cadre (spécifications globales) d'un interface.

2) Lors de la hiérarchisation et de la modularisation, réaliser les implications de ces principes.

Cela entraîne notamment la création de modules de gestion des moyens techniques d'E/S (niveau 4), de modules particuliers liés à la gestion de la commande HELP ou IGNORE,... (voir paragraphe IV.3).

3) Lors de la spécification des modules du niveau 6, adjoindre une spécification, par module, de l'interface.

Cette spécification dont nous venons de préciser la méthode, permettra également d'affiner les spécifications des modules des niveaux 4 et 5 utilisés dans la mise en oeuvre de l'interface.

(C'est le sens de la hiérarchie UTILISE).

En conclusion, on voit donc les liens qui unissent les étapes de conception de l'architecture logique et de l'interface.

V.2 Les principes d'interfaçage.

V.2.1 Les principes.

L'interface homme-machine doit être conçu dans un cadre pré-défini.

Ce cadre permettra en effet d'orienter les spécifications dans un même ordre d'idées.

L'unité conceptuelle ou consistance logique de l'interface, qualité importante, sera ainsi assurée.

Quel est donc pour nous ce cadre, ou encore quels sont les principes d'interfaçage que nous pouvons proposer?

Le principe de facilité d'utilisation (userfriendly) est pour nous essentiel.

En effet, notre logiciel est destiné selon ses objectifs, à des utilisateurs variés et non expérimentés en informatique.

De plus, on peut facilement reconnaître l'accroissement de "rendement" des utilisateurs, même expérimentés, issu de la facilité d'utilisation.

A partir de ce principe de base, nous allons ramener les considérations générale d'interfaçage dans l'optique de sa mise en oeuvre.

V.2.2 La facilité d'utilisation: les règles générales de mise en oeuvre.

Afin de structurer l'énoncé des règles générales d'interfaçage, nous allons reprendre les quatre composants de l'interface

présentés par Newman et Sproull (voir paragraphe I.2.2.2).

Pour chacun de ces composants, nous énoncerons des règles générales d'interfaçage.

V.2.2.1 Au niveau de l'"user's model".

On l'a vu, ce modèle de l'utilisateur reprend les objets (issus de l'application) et les actions sur ceux-ci que l'utilisateur peut réaliser.

A ce propos, nous énonçons les deux règles suivantes :

- . Les objets définis dans le modèle de l'utilisateur doivent être limités dans leur nombre.

Leur signification doit, de plus, être constante tout au long de leur cycle de vie.

Ainsi la notion de surface correspondra à tout moment à un ensemble de points issus d'un phénomène 3-D.

Lorsque nous visualiserons les axes X,Y,Z, ceux-ci ne feront pas partie de l'objet-surface. Ils en seront distincts.

Nous préciserons les objets manipulables dans l'"user's model" (TASK MODEL) des spécifications de l'interface (voir annexe IV).

- . Les actions associées à ces objets seront clairement définies et auront aussi, à la fois, une sémantique et une mise en oeuvre constantes.

A une action seront toujours liées une signification précise et une procédure de mise en oeuvre valable tout au long du logiciel.

Ceci implique d'ailleurs l'existence de séquences fréquentes d'interaction. C'est pourquoi on a créé au niveau 5, un module (n°51) de gestion de ces séquences.

Nous préciserons également ces actions au TASK MODEL de l'annexe IV.

V.2.2.2 Au niveau du "Command Language".

Quel sera le style de langage de commande utilisé?

Nous proposons trois axes pour sous-tendre ce style :

- * 1. Une mise en oeuvre des commandes par menu dont les options sont choisies, du moins à l'écran graphique, par la souris et son curseur.

* 2. Une mise en oeuvre de commandes par des moyens d'input appropriés (boîte à fonctions, table de digitalisation, clavier). Ceci nous a conduit d'ailleurs à créer des modules au niveau 4, de gestion des moyens techniques d'E/S.

* 3. Une mise en oeuvre de moyens d'aide à l'interaction :

- . une commande IGNORE dont le but sera de supprimer l'action réalisée par la dernière commande transmise.
- . une commande HELP dont le but sera de produire des renseignements liés à l'interaction en cours.
- . une mémorisation des commandes appartenant aux fonctions (au sens Bodart) de niveaux supérieurs.

Cette notion de niveau supérieur se place dans le cadre d'une "hiérarchie APPELLE". Ainsi, si une commande C_i située dans la fonction i entraîne l'appel d'une fonction j , i est supérieur à j et C_i sera mémorisée.

- . une présentation de messages de statut invitant l'utilisateur à interagir et explicitant quelque peu la façon de faire (la fonction HELP permettant elle de détailler le mode d'emploi du logiciel).

V.2.2.3 Au niveau de l'"information display".

Nous proposons d'abord de travailler avec deux terminaux (écrans) :

- . l'écran graphique (VG) permettra l'affichage des informations graphiques et de l'ensemble des éléments (menus, messages d'erreur, messages de feedback,...) liés à ces informations et à leur gestion.

- . l'écran annexe (VT) permettra lui l'affichage des informations non-graphiques (données originales, renseignements sur les surfaces,...) et des éléments liés à leur gestion.

Remarquons de suite que le style de commande par menu nécessitera ici un clavier plutôt qu'une souris, si celle-ci n'est pas reliée à l'écran annexe.

Le clavier permettra d'introduire les commandes à choisir au sein du menu.

La raison de l'emploi d'un écran annexe (l'écran graphique étant indispensable pour la visualisation de surfaces 3-D) se trouve principalement dans la difficulté d'afficher des variables de type caractère à l'écran graphique.

Etant donné cette limitation de programmation, il était plus facile pour l'utilisateur de concentrer la gestion des données graphiques sur un même écran graphique et celle des données non graphiques sur un écran terminal classique.

On devra évidemment veiller à ne pas "promener" de façon importante l'utilisateur d'un écran à l'autre. On remarque que ce principe d'interfaçage est fort dépendant du matériel disponible et de la programmation du LGI.

Newman et Sproull ont d'ailleurs soulevé cette dépendance (1979,p448).

Nous proposons ensuite de présenter à l'utilisateur un "look" constant dans sa nature et agréable dans sa présentation pour les deux écrans.

Pour ce faire, nous allons découper les écrans en zones.

A chacune de ces zones seront affectées un type d'information que l'utilisateur pourra y consulter et un type d'information que l'utilisateur pourra y transmettre.

Cette possibilité de transmission d'information est d'ailleurs facilitée par les moyens d'interaction qu'offre le graphique (souris sur écran par exemple).

Nous détaillerons le rôle de ces zones au paragraphe V.2.2.5.

On a compris que ce découpage en zones éliminera les surprises dans les interactions : l'utilisateur pourra prévoir à quel endroit et avec quelle information il pourra traiter tout au long de son dialogue.

V.2.2.4 Au niveau du "feedback".

Toujours dans le cadre d'une facilité à l'usage, l'utilisateur devra être informé du déroulement du programme et de son interaction avec celui-ci.

Newman et Sproull (figure 14) préconisent la technique du feedback à trois niveaux : ."command feedback" : on informe l'utilisateur sur l'acceptation de sa commande, sur l'exécution de celle-ci,

sur l'occurrence d'une condition d'erreur dans le cadre d'une commande.

."selection feedback" : on informe ici, lors de la recherche en B.D d'un élément, si la sélection progresse positivement.

."cursor feedback" : on informe, de façon indépendante à l'application, sur certaines caractéristiques des moyens techniques d'E/S (position de la souris par un curseur, pression d'une touche de clavier par un signal sonore,...).

Il serait intéressant d'implémenter les trois types de feedback.

Remarquons de suite que si le "cursor feedback" semble évident, que si le "command feedback" semble possible à réaliser, le "selection feedback" est plus compliqué à installer.

La B.D doit être organisée de telle manière à pouvoir émettre rapidement le feedback de sélection avant que l'action de sélection ne soit modifiée.

La figure illustre ce problème : les phases (c) et (d) nécessitent une rapidité dans l'adaptation du feedback (inversion noir-blanc d'un texte).

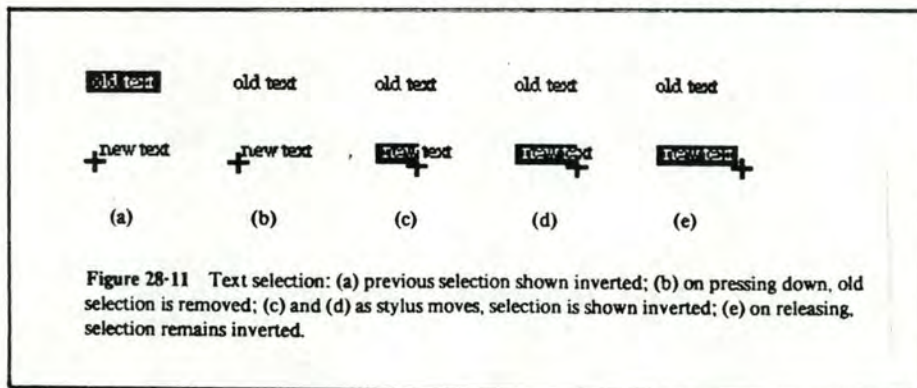


FIGURE 28: Selection feedback : un exemple.

Notons également que ce type de sélection dans la B.D (à partir de l'écran) ne sera pas courant dans notre application : l'accès à la B.D se fera par l'envoi d'un identifiant de l'élément à sélectionner.

A l'écran, la sélection portera donc sur une représentation des données à choisir et non sur des données elles-mêmes.

Par exemple, pour accéder à une surface, on choisira un nom de surface et non un ensemble de points déjà visualisés à l'écran. Le feedback de sélection ne sera donc pas très développé.

En plus de ces trois feedbacks, nous proposons de veiller à informer l'utilisateur sur l'état global d'exécution du logiciel en lui fournissant :

- les références des données sur lesquelles il travaille
- les références de la phase et de la fonction où il se trouve.

V.2.2.5 Généralités sur l'utilisation de l'écran graphique et annexe.

On trouvera à la figure 29 la description schématique des zones des deux écrans. Voyons leur rôle spécifique.

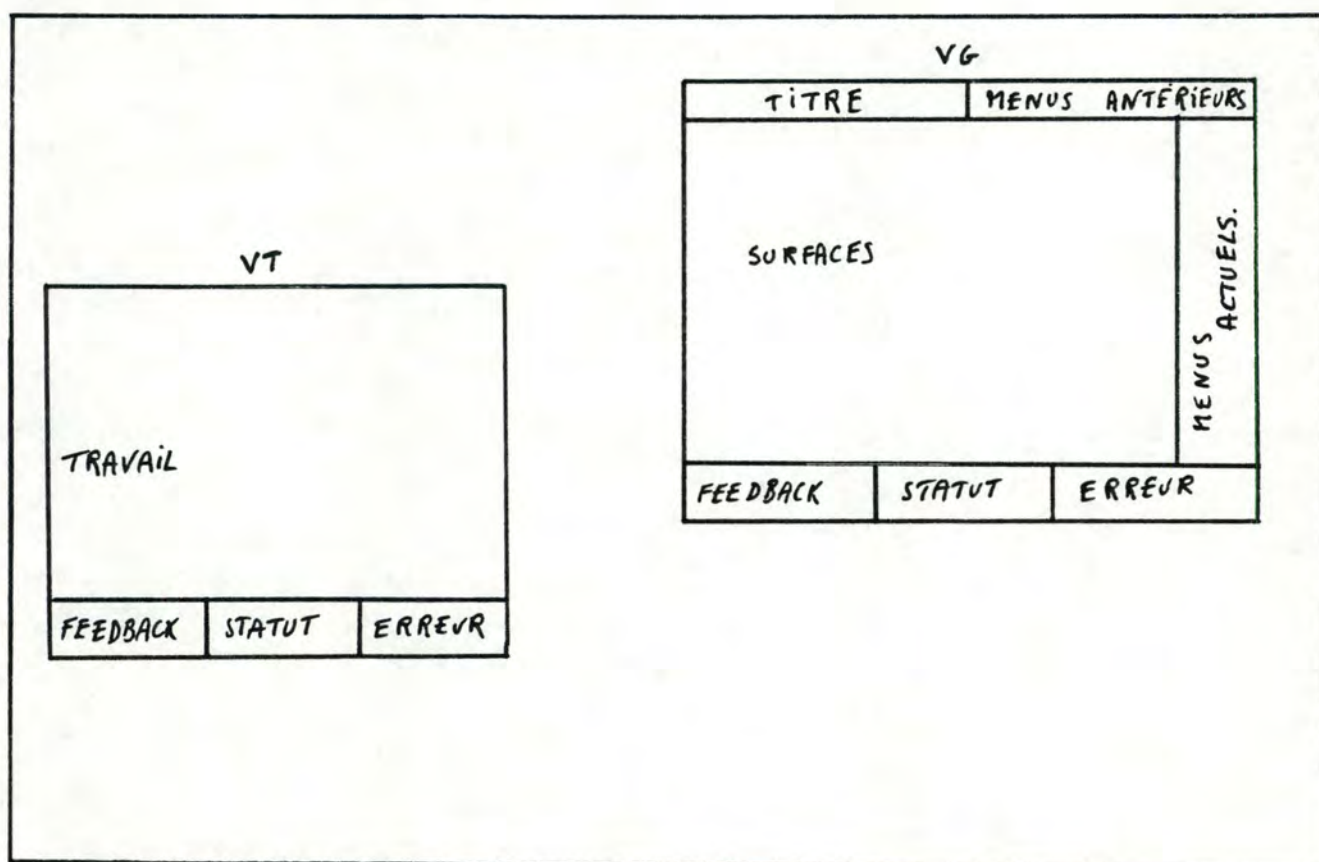


FIGURE 29 : Les zones des écrans graphique et annexe.

Pour l'écran graphique:

. Zone de TITRE : En sortie, l'utilisateur trouvera le nom de la phase et de la fonction correspondant à l'état actuel d'exécution du logiciel. Il y trouvera également le nom des surfaces affichées dans la zone des SURFACES. Ces noms de surfaces seront visualisées avec la même intensité que les surfaces.

. Zone des SURFACES : En sortie, les surfaces à l'étude seront visualisées. L'utilisateur pourra, en "entrée", sélectionner des éléments de cette surface (point).
On parlera de pseudo-entrée ("entrée") car l'écran doit être accompagné d'un crayon lumineux ou d'une souris pour réaliser l'entrée d'une information.

. Zone du MENU ACTUEL : En sortie, les commandes actuellement disponibles seront spécifiées par une abréviation significative. La commande actuellement en exécution sera affichée en inversion (fond lumineux, texte noir).

En "entrée", l'utilisateur pointera une zone d'item du menu actuel, du moins si celle-ci est activable, par la souris.

La souris ne pourra donc se déplacer que dans les zones activables du menu actuel.

. Zone des MENUS ANTERIEURS : En sortie, cette zone reprendra les commandes de niveaux supérieurs (en termes d'appel de fonctions ou de phases) choisies auparavant par l'utilisateur.

. Zone de STATUT : En sortie, on y trouvera des messages décrivant^{ce} que le logiciel attend de l'utilisateur dans l'interaction suivante (exemple : "Voulez-vous choisir une option au menu actuel?")

. Zone de FEEDBACK : En sortie, on y trouvera les messages porteurs de feedback. Ils correspondront principalement à l'état, positif ou négatif, d'avancement d'une commande (exemple : "OK, je calcule").

. Zone des ERREURS : En sortie, on y trouvera les messages d'erreur explicitant une progression négative d'une commande, ses raisons et ses solutions (exemple : "commande invalide, veuillez choisir une autre option").

Pour l'écran annexe :

Nous devons tenir compte de la possibilité ou non de gérer l'écran par un gérant d'écran, qui serait alors un outil supplémentaire du niveau 2 de notre architecture.

Nous devons également remarquer l'impossibilité de brancher une souris sur un écran annexe classique.

L'absence éventuelle d'un gérant d'écran rendra la tâche difficile dans la gestion des différentes zones.

Deux possibilités s'offrent donc à nous.

Si l'on dispose d'un gérant d'écran, on peut spécifier les mêmes zones que pour l'écran graphique si ce n'est que

- . la zone des surfaces sera dénommée zone de TRAVAIL et ne pourra recevoir en sortie que des informations non-graphiques.
- . la zone du MENU ACTUEL ne sera pas pointée par un curseur lié à une souris.

On entrera au clavier les commandes inscrites dans ce menu actuel.

Ceci a le désavantage de modifier le style de commande (du menu avec souris, on passe au menu avec clavier).

Si l'on ne dispose pas d'un gérant d'écran, on pourrait délivrer et recevoir les informations liées à la gestion des données non-graphiques, à l'écran graphique.

L'écran annexe ne comportant alors qu'une zone de travail.

Ceci n'est pas très aisé pour l'utilisateur dont par exemple les messages d'erreur lui apparaîtront à un autre écran qu'à celui auquel il est attaché au travail.

Le moyen terme choisi est :

- . d'implémenter, concernant la gestion des données non-graphiques, les zones de FEEDBACK, STATUT et ERREUR à l'écran annexe.

- . de conserver, afin de préserver l'unité de style de commande, une gestion des commandes réalisée à partir de l'écran graphique et de ses zones de MENU ACTUEL et MENUS ANTERIEURS.

- . de rappeler, si certaines actions d'interactions nécessitent absolument des commandes au clavier, la signification de ces commandes dans la zone de STATUT de l'écran annexe.

V.3 Spécification de l'interface.

On trouvera en annexe IV la spécification, du moins pour certains modules de base du niveau 6, de l'interface et ceci, selon la méthode rappelée au paragraphe V.1.

V.4 Conclusion.

Ce chapitre représente, nous semble-t-il, un pas important dans la réalisation d'un logiciel dont la procédure de conception tient compte des particularités de celui-ci.

L'interface est en effet un élément essentiel d'un LGI.

Bien le concevoir est donc primordial.

De plus, nous avons tenté d'intégrer cette conception dans une démarche classique globale.

Cette intégration nous montre deux interdépendances notables :

- . la spécification de l'interface est liée à la modularisation. Cette dernière contribue en effet à mettre en oeuvre les spécifications de l'interface. On l'a vu au chapitre IV.

- . la spécification de l'interface doit se préoccuper des moyens techniques d'E/S disponibles.

En effet, le langage de commande de l'interface sera tributaire des "devices" disponibles et de leurs caractéristiques.

CHAPITRE VI

CONCEPTION ET MISE EN OEUVRE PHYSIQUE DES MODULES.

VI.1 Rappels méthodologiques.

Deux étapes distinctes sont à réaliser dans ce chapitre :

- . L'élaboration des algorithmes des modules de l'architecture logique.

- . La mise en oeuvre physique des modules qui comprend leur découpe en éléments exécutables et le codage de ces éléments. Ces éléments de programmation formeront l'architecture physique du logiciel. En général, un module génère un élément exécutable.

Ces deux étapes furent regroupées au sein de ce même chapitre. Elles conduisent en effet toutes deux à la concrétisation des spécifications abstraites (Annexe III) et de l'interface (Annexe IV).

A ce niveau de l'application de notre démarche, nous allons procéder à une troisième restriction d^e plan de travail initial. Cette restriction suggère de choisir parmi les 14 modules spécifiés en annexe III, cinq modules (n°69,53,410,412 et 413) pour les étapes finales de conception algorithmique, et si nous en avons le temps, de codage et de tests.

Ces cinq modules, malgré leur nombre restreint, permettront, à partir des données originales d'une surface, d'en produire une visualisation à l'écran.

Loin de nos objectifs de base, ce "core" modulaire pourra mettre en évidence la faisabilité des ultimes étapes de notre démarche.

VI.2 Algorithmes de conception.

Nous présentons en annexe V les algorithmes écrits en un pseudo-langage proche des structures du PASCAL.

VI.3 Elaboration d'une architecture physique : les choix préalables d'implémentation physique.

Une architecture physique correspond à un ensemble d'éléments exécutables reliés par des relations physiques (appel, réveil,...).

Ces éléments exécutables (routines, procédures, sous-programmes, programmes) sont issus de l'architecture logique.

Au préalable, nous devons effectuer certains choix d'implémentation physique concernant :

- . le matériel utilisé pour implanter le logiciel.
- . le LGDB du niveau 2 de l'architecture logique.
- . le langage de programmation.

VI.3.1 Le système Megatek.

La configuration générale du système graphique basé aux Facultés Notre-Dame de la Paix est décrite à la figure 30.

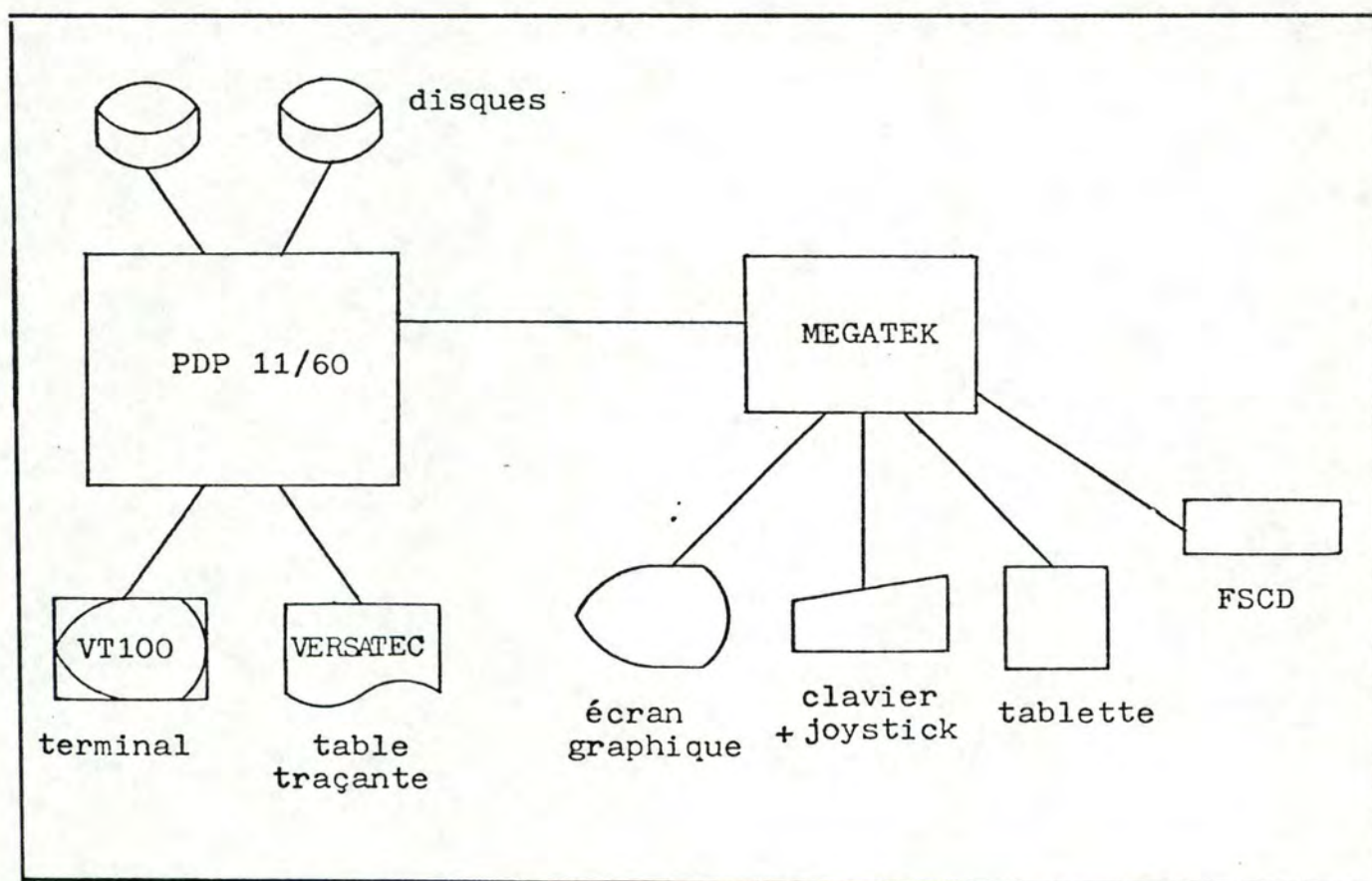


FIGURE 30: Configuration du système graphique de Namur.

Le PDP 11/60 gère le système Megatek (type WHIZZARD 7200) par le biais du système d'exploitation RSX11-M (Version 3.2).

Précisons maintenant la configuration du système Megatek. La figure³¹ décrit l'architecture fonctionnelle du système WHIZZARD 7200 de Megatek à écran vectoriel ("Line-drawing"; voir figure 5).

Figure³¹ : voir page suivante.

Pour l'"output process", on y observe les éléments suivants : a) INTERFACE : Ce module permet la communication à la mémoire graphique (display list memory) des instructions de lecture et écriture mais aussi l'entrée des informations à partir des périphériques, vers le programme d'application.

b) DISPLAY LIST MEMORY : Cette mémoire contient des instructions capables de générer les déplacements du vecteur de l'écran (voir figure 6), mais aussi de positionner les variables telles l'intensité, la couleur, la forme des traits. Cet ensemble d'instructions forment réellement un programme graphique dont une modification quelconque implique un changement de l'image à l'écran.

Les instructions de la liste d'affichage (display list memory) sont organisées en segments auxquels certains attributs seront affectables (visibilité,...).

c) GRAPHICS PROCESSOR : Ce processeur lit et interprète les instructions de la liste d'affichage. Il transmet le résultat de son interprétation au module suivant. Ce processeur doit être suffisamment rapide pour assurer le maintien de l'image.

d) HCRST MULTIPLIER : Ce module est un multiplicateur hardware de matrices (Hardware Clip Rotate Scale and Translate). Il transforme rapidement les vecteurs des segments de la liste d'affichage. Il permet la rotation, le changement d'échelle et la translation de façon hardware en libérant l'ordinateur-hôte de ces tâches et en accroissant le taux d'interactivité.

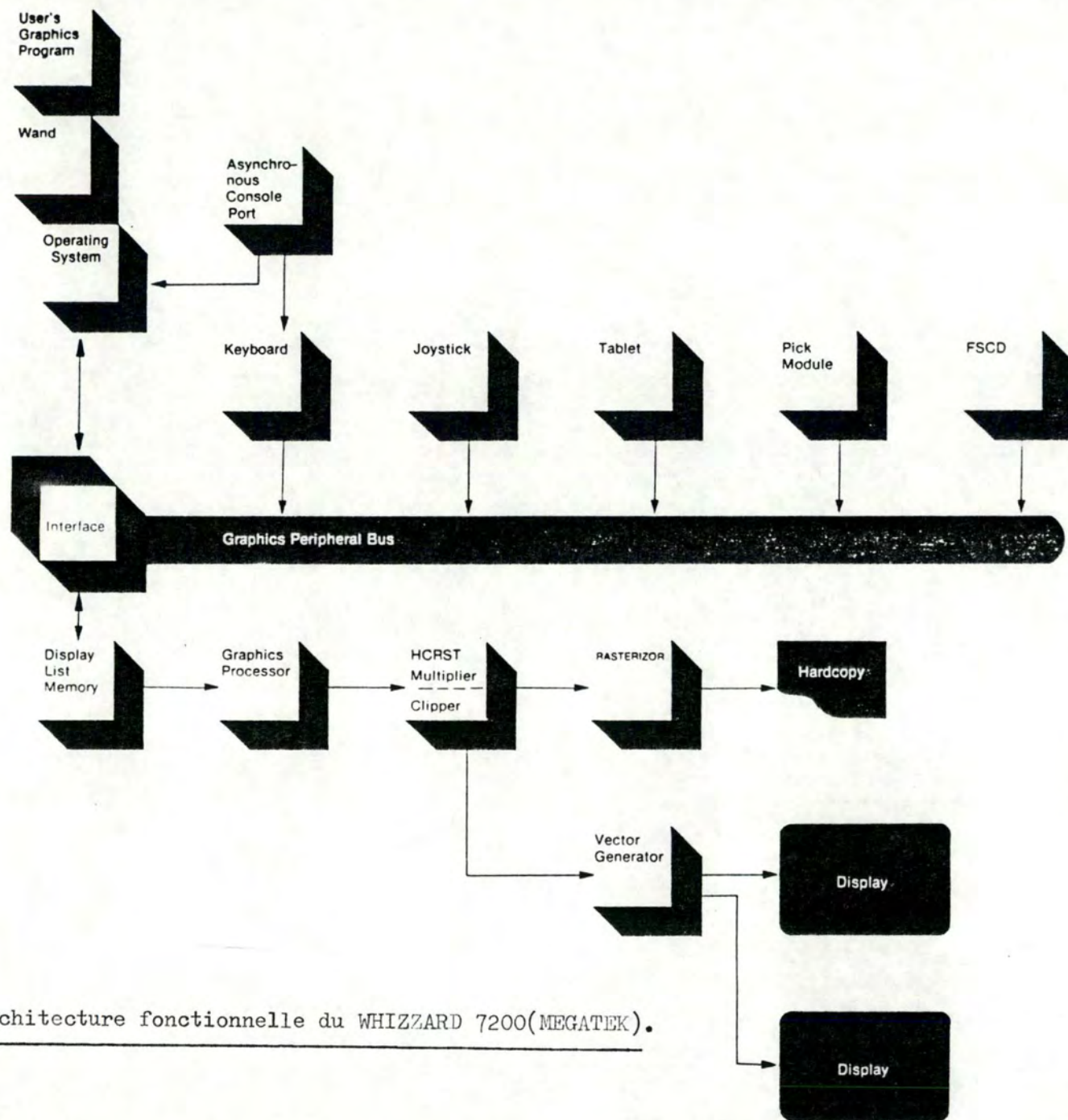


FIGURE 31 : Architecture fonctionnelle du WHIZZARD 7200(MEGATEK).

e) HCRST CLIPPER : Ce module permet la définition d'un champ de visualisation sur l'écran (voir figure 13), adapté aux besoins de l'utilisateur.

Le terme clipper est nous semble-t-il, mal choisi, car relié à une autre opération, réalisée elle, dans l'espace des coordonnées-objet afin de supprimer les parties invisibles de cet objet (clipping).

f) VECTOR GENERATOR : En fin de processus de visualisation, le générateur de vecteurs produit à l'écran les traits, et ce, selon la liste d'affichage. Il convertit donc l'information digitale en sa représentation graphique.

La communication entre ces modules de visualisation se fait via le bus graphique.

Concernant l'"input process", on dispose des périphériques suivants, connectés au bus des périphériques : un clavier, un joystick, une table de digitalisation, un gestionnaire d'un crayon lumineux (Pick module) et un gestionnaire de boîte à fonctions et potentiomètres (FSCD).

La figure³¹ illustre également le rôle de tampon du LGDB WAND entre le système d'exploitation (niveau 1 de l'architecture) et le programme d'application (niveaux supérieurs).

VI.3.2 Le LGDB WAND 7200.

Adapté au système WHIZZARD 7200, le LGDB proposé par Megatek est le WAND 7200 (1981).

Ecrit en FORTRAN, ce logiciel graphique de base répond aux caractéristiques générales suivantes :

- . compatibilité^a avec des écrans graphiques "line-drawing" (vecteur) et "storage" (raster).
- . compatibilité avec les essais de normalisation du groupe SIGGRAPH d'ACM.
- . mise en oeuvre des techniques de segmentation de la liste d'affichage.
- . mise en oeuvre de transformations hardware via le HCRST.
- . mise en oeuvre des périphériques graphiques classiques.

Les fonctionnalités du WAND 7200 sont :

- . gestion de la station de travail (connexion,...)

- . primitives de base de génération d'un vecteur (MOVE, DRAW,...)
- . primitives de segmentation de la liste d'affichage.
- . primitives de paramétrisation des attributs d'un vecteur (couleur, intensité, forme,...).
- . primitives de paramétrisation des attributs d'un segment (visibilité,...).
- . primitives de gestion des entrées par les différents périphériques.
- . primitives de transformation (rotation, translation, scaling, clipping) en 2-D et en 3-D.

On remarque que ces primitives sont très proches de celles énoncées dans le cadre de l'étude de Newman et Sproull (1979, voir paragraphe I.2.2.1.2).

Le LGDB peut donc répondre à nos attentes.

VI.3.3 Le langage de programmation.

Nous avons choisi le FORTRAN 77 pour langage de programmation.

Ce langage fait l'unanimité dans le domaine des applications graphiques qui comportent une part importante d'algorithmes scientifiques.

Afin de faciliter les aspects "gestion" (fichiers,...), nous avons choisi la version 77 du FORTRAN.

CONCLUSIONS GENERALES

"Comment concevoir un logiciel graphique interactif?"

Telle était la première question centrale de notre recherche.

La réponse à cette question passait par la mise sur pied d'une méthodologie conforme à trois critères.

Ces trois critères, rappelons-les, invoquent :

- . le caractère d'optimisation des qualités du logiciel (fiabilité, maintenance, portabilité, efficacité, userfriendly, sécurité),
- . le caractère opérationnel de la méthode,
- . le caractère d'adaptation de la méthode aux caractéristiques d'un logiciel graphique interactif.

Ces trois mêmes critères vont être utilisés pour évaluer les résultats de nos investigations.

La méthode proposée se base sur les théories de Bodart (1983) et Van Lamsweerde (1983).

Ces théories classiques de développement de logiciels sont synthétisées à la figure 10.

Remarquons tout d'abord que la démarche issue de ces théories classiques subira quelques restrictions dans ses premières étapes. En effet, notre travail ne se base pas sur un existant comme l'envisage Bodart dans son étude d'opportunité.

Ainsi, l'étude d'opportunité et notamment son analyse de l'existant et sa critique de l'existant sont inutiles dans notre cas.

On s'est donc, dans cette étude d'opportunité, limité à préciser les spécifications globales (requirements) du squelette de la solution à créer.

L'originalité de nos résultats est d'intégrer au sein de la démarche classique des éléments susceptibles de tenir compte des particularités d'un logiciel graphique interactif.

Celles-ci sont au nombre de deux.

La diversité des moyens techniques d'E/S est la première. Elle a induit la mise en oeuvre d'un logiciel graphique de base afin de gérer ces moyens techniques d'E/S de façon optimale et indépen-

dante vis-à-vis du matériel d'E/S.

Les efforts de standardisation des LGDB sont dans ce sens, essentiels.

La seconde particularité est l'importance de l'interface homme-machine.

L'interactivité d'un logiciel graphique ne peut donner toute sa mesure que si elle se déroule dans de bonnes conditions pour l'utilisateur.

On se doit donc de prêter attention à celles-ci.

Quels sont les éléments méthodologiques qui nous ont permis de prendre en compte ces deux particularités?

Tout d'abord, au sein de l'architecture modulaire issue de l'emploi de la hiérarchie UTILISE, nous proposons au niveau 2, d'insérer un LGDB dont les primitives permettront une gestion optimale des moyens techniques d'E/S.

Ensuite, nous avons proposé de spécifier l'interface homme-machine. Cela peut se faire en trois étapes :

- définir les principes généraux d'interfaçage.
- préciser lors de la modularisation de l'architecture logique, les modules nécessaires à l'application de ces principes.
- spécifier l'interface, selon les méthodes de Green et Newman et Sproull, au niveau 6 de l'architecture.

La notion d'"abstraction utilisateur" est ainsi intégrée dans les concepts de spécifications abstraites de modules.

Afin d'évaluer cette méthodologie, nous allons reprendre les trois critères guides de notre recherche et répondre à la question suivante : "Ces critères furent-ils mis en oeuvre de façon optimale?"

Etant en parfait accord avec la démarche classique (Bodart et Van Lamsweerde), la démarche proposée permet d'atteindre les qualités attendues de tout logiciel.

La démarche tient compte de façon évidente des particularités d'un LGI. Elle s'y prête d'ailleurs bien.

Ainsi, le concept de LGDB, grâce à la structure hiérarchique, s'intègre parfaitement dans l'ensemble de notre méthodologie.

La démarche est-elle opérationnelle?

A cette question, nous devons répondre de façon nuancée.

En effet, sous ce point de vue, il n'est possible d'évaluer nos résultats qu'après application complète de la démarche à un cas concret.

La complétude n'a pu être atteinte faute de temps.

Mais, et ce de façon encourageante, un stade avancé d'application a pu être dégagé tout en suivant de façon scrupuleuse, la méthode proposée.

Nous détaillerons ce stade avancé lors de l'évaluation de l'application de la méthode.

Le second objectif du mémoire était l'application de notre démarche de conception afin de produire un logiciel de manipulation interactive de surfaces 3-D.

Les objectifs du logiciel étaient exigeants et ceci dans un souci de qualité du produit.

En plus des objectifs fonctionnels (saisie, visualisation, manipulation de surfaces 3-D quelconques), nous avons énoncé des objectifs au niveau des moyens (graphiques), des qualités attendues et des utilisateurs (non expérimentés et diversifiés).

Avant d'évaluer le produit logiciel obtenu, nous devons examiner les conditions d'application de la démarche de conception proposée.

Nous posons au départ les questions suivantes : la démarche est-elle opérationnelle dans le cadre de notre application, la démarche présente-t-elle des intérêts?

Notre réponse à la première est globalement positive.

Nous avons en effet, pour certains modules de l'architecture logique, pu atteindre l'étape de conception algorithmique.

Nous ne pouvons évidemment statuer définitivement sur ce caractère "opérationnel" de la démarche, tant que le produit logiciel ne "tournera" pas.

Mais il est certain que la méthode proposée est possible, et dans de bonnes conditions.

La continuation du travail devrait le démontrer définitivement.

A la seconde question, notre recherche permet d'y répondre favorablement.

La démarche intègre, on l'a vu, positivement les particularités d'un LGI et implique une spécification complète de tous les éléments de ce LGI, y compris son interface.

La démarche permettrait une automatisation de certaines de ses étapes (analyse fonctionnelle, plans de test).

L'application scrupuleuse de la démarche nous a permis, jusqu'à l'étape de conception algorithmique, d'éviter les erreurs cruciales qui impliquent une refonte de tout l'acquis des étapes antérieures à l'étape où l'erreur fut détectée.

L'extensibilité du produit nous semble possible.

L'architecture modulaire y contribue.

Ainsi, si l'on veut introduire des formes d'équation supplémentaires, il suffira de modifier les modules d'introduction des données (n°617), d'enregistrement de celles-ci (n°41) et de calcul des coordonnées-écran (n°53).

De même, si l'on désire adjoindre un type supplémentaire de paramètre de visualisation, on modifiera le module de gestion (n°610) et de mise en oeuvre (n°53) de ce type.

Deux difficultés sont à noter dans l'application de la démarche.

La première est issue de la complexité du graphique. Certains problèmes (Vu et caché, optimisation de la représentation en machine, optimisation de l'usage des primitives du LGDB, conception de l'interface,...) sont essentiels et doivent être appréhendés et même plus ou moins résolus avant l'étape de modularisation de l'architecture logicielle.

Un retard évident de cette modularisation fut donc inévitable.

La seconde difficulté provient de la quantité de travail engendrée par l'application scrupuleuse de la méthode. Cependant, nous croyons en son utilité pour assurer la valeur des résultats même partiels.

Ces deux difficultés ont donc fortement influencé l'état d'avancement du produit logiciel.

A notre regret, nous n'avons pu réaliser dans les délais imposés, un produit logiciel final. Et ce, malgré la stratégie de restriction choisie pour mener notre plan de travail.

Trois restrictions furent décidées :

- . limitation des surfaces 3-D à étudier.
- . limitation des spécifications abstraites et d'interface à une quinzaine de modules, essentiels à la mise en oeuvre des objectifs de base.
- . limitation de la conception algorithmique à quatre modules essentiels à la visualisation d'une surface.

Toutefois, les résultats de conception obtenus permettent, nous semble-t-il, d'assurer la possibilité d'une implémentation complète du logiciel.

De façon intégrale, les étapes de spécifications globales, d'analyse fonctionnelle, d'hiérarchisation, de modularisation furent établies. Les spécifications abstraites, la spécification de l'interface, la conception algorithmique furent initialisées.

L'accord du produit avec une démarche solide nous fait croire à la qualité des résultats partiels obtenus.

BIBLIOGRAPHIE

BIBLIOGRAPHIE.

- AKIMA H, 1978, "A method of bivariate interpolation and smooth surface fitting for irregularly distributed data points", A.C.M Transactions on Math. Software, 4, n°2.
- AMSON J-C, 1972, "Equilibrium models of cities : 1. An Axiomatic Theory", Environment and Planning, 4, p429-444.
- BEGUIN H, 1979, Méthodes d'analyse géographique quantitative, LITEC, Paris.
- BERTIN J, 1973, Sémiologie graphique, Mouton, Gauthier-Villars, Paris
- BODART F, 1983, Cours d'analyse fonctionnelle, F.U. Namur
- BRUNET P, 1983, "On surface representation from irregularly distributed data points", Technique et Science informatiques, Vol 2, n°2, p 103-110.
- BUSSIERE R, STOVALL T, 1981, Systèmes évolutifs urbains et régionaux à l'état d'équilibre, Centre de recherche et de rencontres d'urbanisme, Paris.
- CONNOLLY M, 1982, "User's guide to Molecular Surfaces", Department of Molecular Biophysics and Biochemistry, Yale University, New Haven, U.S.A.
- DELFINER P, DELHOMME J-P, 1975, "Optimum interpolation by Kriging" in DAVIS, MC CULLAGH, Display and analysis of spatial data, NATO, Wiley, Londres.
- FOLEY J-D, VAN DAM A, 1982, Fundamentals of interactive computer graphics, Addison Wesley, Reading, U.S.A.
- GKS, 1979, Information Processing Graphical Kernel System (GKS), Proposal of standard, DIN 0066252.
- GREEN M, 1981 a, "A specification language and design notation for graphical user interface", TR 81-CS-09, MacMaster University.

- GREEN M, 1981, "A methodology for the specification of Graphical User interface", A.C.M Computer Graphics, Vol 15, n°3, p 99-108.
- GSPC SIGGRAPH, 1977, General Methodology and proposed Standard, A.C.M. Computer graphics, Vol 11, n°3.
- HAYNES K.E, RUBE M.I, 1973, "Directionnal bias in urban population density", Amals of American Association of geographers, 63, n°1, p 40-47.
- JOURET B, 1976, "Méthode d'analyse géographique de l'espace et techniques informatiques. Application à Bruxelles", Bul.de la Soc. belge de photogrammétrie, n°21, p 41-61.
- LUCAS M , 1982, La réalisation des logiciels graphiques interactifs, Eyrolles, n°41, Paris.
- MORVAN P, LUCAS M, 1976, Images et ordinateur-Introduction à l'infographie interactive, Larousse, Série informatique, Paris.
- NEWMAN W.M, SPROULL R.F, 1979, Principles of interactive Computer Graphics, 2nd edition, Mac Graw Hill, New-York.
- NEWLING B, 1969, " The Spatial Variation of Urban Population densities", Geographical review, 59, p 242-52.
- SPRUNT B.F, 1975, "Relief representation in automated Cartography : an algorithmic approach", in Display and Analysis of spatial data, (Voir Delfiner et Delhomme).
- ROGERS D.F, ADAMS J.A, 1976, Mathematical Elements for Computer graphics, Mac Graw Hill, New-York.
- VAN LAMSWEERDE A, 1983, Cours de méthodologie de développement de logiciels, F.U. Namur.
- WAND, 1981, Introduction to Wand 7200, Megatek Corporation, San Diego, U.S.A.

WU SC, ABEL J-F, GREENBERG D.P, 1977, "An interactive Computer Graphics
Approach to surface representation",
A.C.M Communications, Vol 20, n°10,
p 703-712.

ZIELENSKI K, "Experimental analysis of eleven models of urban population
density", Environment and Planning, 1979, p 629-41.

ANNEXES

ANNEXE O

LISTE DES FIGURES.

<u>Figure n°</u>	<u>Intitulé</u>	<u>Page</u>
1	Deux types de surfaces de dessin.	6.
2	Un modèle simple du processus d'interaction.	8.
3	Un modèle simple du processus d'interaction.	9.
4	Un modèle simple du processus de visualisation sur surface à trait.	10.
5	Schéma général de fonctionnement d'un écran "Line-drawing".	10.
6	Instruction graphique simplifiée.	11.
7	Un modèle simple de processus de visualisation sur surface point à point.	11.
8	Un modèle simple de processus d'entrée.	12.
9	Output-process avec utilisation d'un graphics package.	16.
10	Un méta-algorithme de conception de logiciel.	18.
11	Hiérarchisation type d'un système.	19.
12	Schéma de modularisation des LGI avec LGDB.	22.
13	Notion de fenêtre et de champ de visualisation:schéma.	24.
14	Types de feedbacks.	25.
15	Comparaison entre les méthodes de Green et Newman et Sproull.	26.
16	Classification des surfaces 3-D.	34.
17	Les fonctions de base du logiciel général.	35.
18	Visualisation en "Wire-frame".	41.
19	Un système d'axes Left-handed.	41.
20	Le clipping.	42.
21	Les fonctions de base du logiciel appliqué à la géographie.	47.
22	Surface Contact - Surface Réentrant.	53.
23	L'analyse fonctionnelle et ses résultats.	55.
24	Dynamique des phases.	60.
25	La dynamique liée à la phase d'aide.	61.
26	Schéma global des données (E/A).	63.
27	Les types de surface et leur cycle de vie.	71.
28	Sélection feedback : un exemple.	83.
29	Les zones des écrans graphique et annexe.	84.
30	Configuration du système graphique de Namur.	89.
31	Architecture fonctionnelle du Whizzard 7200(Mégatek)	91.
32	Projection perspective.	210.
33	Pyramide de visualisation du clipping.	211.

LISTE DES TABLEAUX. (en annexe 0).

<u>Tableau n°</u>	<u>Intitulé</u>
1	Liste des modules du niveau 6.
2	Découpe en fonctions de l'application.
3	Liste des modules du niveau 5.
4	Liste des modules du niveau 4.

TABLEAU 1 : Liste des Modules du Niveau 6.

N°	NOM DU MODULE	N° DE LA FONCTION	
61	INITIALISATIONS SESSION	1 de la phase 1	
62	INITIALISATIONS SYSTEME	2 de la phase 1	MV2
63	CHOIX SURFACE et INIT.OPERATION-VERS.SIMPLE	1,2 de la phase 2A	MV1
64	CHARGEMENT DONNEES SURFACE ORIGINALE	2 des phases 2A,B,C	MV2
65	CHOIX SURFACE et INIT.OPERATION-VERS.CHIMIQUE	1,3 de la phase 2C	MV1
66	CHOIX SURFACE et INIT.OPERATION-VERS.GEOGRAPH.	1,3,4 de la phase 2B	MV1
67	CALCUL MODELISATION	1 de la phase 9	MV1
68	CALCUL RESIDUS	2 de la phase 9	MV1
69	INITIALISATIONS ETUDE 1 SURFACE	1 de la phase 3	
610	GESTION VISUALISATION SURFACE	3 de la phase 4 et	MV2
		3 de la phase 3	
611	GESTION MANIPULATION SURFACE	4 de la phase 4 et	
		3 de la phase 3	
612	INITIALISATIONS ETUDE 2 SURFACES	1 de la phase 4	
613	GESTION COMPARAISON SPATIALE	2 de la phase 4	
614	GESTION COMPARAISON STATISTIQUE	5 de la phase 4	
615	ENREGISTREMENT RENSEIGNEMENT SURFACE	1 de la phase 5	MV2
616	ENREGISTREMENT DONNEES SURFACE P	2 de la phase 5	MV2
617	ENREGISTREMENT DONNEES SURFACE E	2 de la phase 5	MV2
618	INITIALISATION MISE A JOUR SURFACE	3 de la phase 5	MV2
619	MISE A JOUR RENSEIGNEMENT SURFACE	4 de la phase 5	MV2
620	MISE A JOUR DONNEES SURFACE	5 de la phase 5	MV2
621	INITIALISATION MODIFICATION REPRESENTATION	1 de la phase 6	
622	CALCUL MODIFICATION REPRESENTATION	2 de la phase 6	
623	TERMINAISON SESSION	1 de la phase 8	
624	TERMINAISON CONNEXION	2 de la phase 8	

TABLEAU 2 : Découpe en fonctions de l'application.

N° PHASE	N° FONCTION	NOM DES FONCTIONS
1	1	Introduction et tests des caractéristiques d'exécution.
	2	Initialisations du système.
2A	1	Choix des surfaces à opérer.
	2	Chargement des données de la surface originale.
	3	Initialisation de l'opération sur une surface originale.
2B	1	Choix des surfaces géographiques à opérer.
	2	Chargement des données de la surface originale géographique.
	3	Initialisation de l'opération sur une surface originale géographique non modélisée.
	4	Initialisation de l'opération sur une surface originale géographique modélisée.
2C	1	Choix des surfaces chimiques à opérer.
	2	Chargement des données de la surface originale chimique.
	3	Initialisation de l'opération sur une surface originale chimique.
3	1	Gestion de l'affichage de la surface à l'étude et coordination.
	2	Gestion des caractéristiques de la visualisation d'une surface visualisée.
	3	Gestion des manipulations de la surface visualisée.
4	1	Gestion de l'affichage de deux surfaces à l'étude et coordination.
	2	Gestion des caractéristiques de visualisation d'une surface visualisée.
	3	Gestion des manipulations d'une surface visualisée.
	4	Gestion de la comparaison spatiale de 2 surfaces visualisées.
	5	Gestion de la comparaison statistique de 2 surfaces visualisées.

TABLEAU 2 : Suite.

N° PHASE	N° FONCTION	NOM DES FONCTIONS
5	1	Introduction des renseignements généraux d'une surface.
	2	Introduction des données d'une surface.
	3	Coordination des mises à jour d'une surface.
	4	Mise à jour des renseignements généraux d'une surface.
	5	Mise à jour des données d'une surface.
6	1	Gestion de la mise en oeuvre de la modification de représentation.
	2	Calcul des traitements d'approximation et d'interpolation locale.
7	1	Gestion de l'appel HELP.
	2	Gestion de la documentation.
8	1	Terminaison d'une session.
	2	Terminaison d'une connexion au logiciel.
9B	1	Traitements complémentaires à une modélisation.
	2	Traitements complémentaires à un calcul de résidus.

TABLEAU 3 : Liste des modules de niveau 5.

N°	NOM du MODULE
51	GESTION SEQUENCES D'INTERFACE
52	CALCUL VU ET CACHE
53	CALCUL COORD. ECRAN
54	GESTION HISTORIQUE SURFACE
55	GESTION HISTORIQUE COMMANDES
56	GESTION AIDE UTILISATEUR
57	GESTION MEMORISATION SURFACE.

TABLEAU 4 : Liste des modules du niveau 4.

N°	NOM du MODULE		
41	SURFACE ORIGINALE	MV2	MD
42	MANIPULATION	MV2	MD
43	SURFACE MODELISEE GEOGRAPHIQUE	MV1	MD
44	SAISIE et VALIDATION SYNTAXIQUE		
45	VALIDATION SEMANTIQUE		
46	GESTION ZONE FEEDBACK		
47	GESTION ZONE STATUT		
48	GESTION ZONE TITRE		
49	GESTION ZONE ERREUR		
410	GESTION ZONE MENU ACTUEL		
411	GESTION ZONE MENU ANTERIEUR		
412	GESTION SOURIS		
413	GESTION ZONE SURFACE		
414	GESTION BOITE FONCTIONS		
415	GESTION TABLE		
416	AFFICHAGE RESULTATS		
417	DOCUMENTATION		MD
418	ORGANISATION SYSTEME		MD

ANNEXE I.

Objectifs et Contraintes des Phases de l'application.

A I 1 Phase n°1 : Initialisation d'une session.

A I.1.1. Objectifs.

- . Assurer les initialisations nécessaires au système
 - interactives : on identifiera l'utilisateur dans le but d'une sécurité d'accès au système et ce par rapport à une version choisie; on testera ensuite la validité des ressources de traitement disponibles en fonction du choix de version.
 - non-interactives: on ouvrira les fichiers de la B.D; on initialisera les variables du système.
- . Assurer en fonction des initialisations (utilisateur autorisé;ressources de traitements,valides,de la version choisie),l'entrée réelle dans le système ou sa sortie.
- . Assurer,pour l'utilisateur privilégié,la mise à jour des listes d'utilisateurs et des ressources du logiciel.

A I.1.2. Contraintes des Performances.

A I.1.2.1. Contraintes de base.

Les contraintes seront reprises au sein de la plupart des phases et ont donc reçu le qualificatif de base.On peut citer:

- . Clarté du dialogue pour les éléments interactifs de la phase.
- . Rapidité des traitements pour les éléments non interactifs de la phase.
- . Temps de réponse dans tous les cas inférieurs à 5 secondes.

A I.1.2.2. Contraintes particulières.

- . Un utilisateur pourra travailler au sein de plusieurs versions,mais dans au moins une.
- . Il devra y avoir au moins un utilisateur privilégié.
- . L'identification d'un utilisateur se fera par son NOM et son MOT de PASSE.

A I.2. Phase n°2A : Coordination - Version Simple.

A I.2.1. Objectifs.

- . Assurer la coordination des opérations (au sens défini au paragraphe III.2.2.), sur une surface,
 - en permettant l'introduction et le chargement des données des surfaces sur lesquelles l'utilisateur opère,
 - en aiguillant l'utilisateur dans son choix d'opération et de surface visée par l'opération,
 - en assurant un point de retour après terminaison de l'opération.
- . Assurer la fin de l'exécution de la phase de coordination dans le but d'une terminaison de la session.

A I.2.2. Contraintes de Performances.

A I.2.2.1. Contraintes de base.

Voir paragraphe A I.1.2.1.

A I.2.2.2. Contraintes particulières.

Cette phase est centrale pour l'utilisateur. On y peaufinera donc la qualité du dialogue.

Cette phase est le préliminaire avant le déclenchement d'une opération sur une surface, notamment une étude (au sens défini au paragraphe III.2.2.) interactive. On y concentrera donc les traitements lents de chargement des données de la surface, choisie pour objet d'opération.

A I.3 Phase n°2B : Coordination - Version géographique.

A I.3.1. Objectifs

Dans le cadre de l'application géographique, on veut par cette phase :

- . Assurer la coordination des opérations sur une surface géographique,
 - en permettant l'introduction et le chargement des données de surfaces sur lesquelles l'utilisateur opère,
 - en aiguillant l'utilisateur dans son choix d'opération (y compris la modélisation interactive) et de surface (y compris les surfaces géographiques modélisées) visée par l'opération.
- . Assurer la fin d'exécution de la phase de coordination dans le but d'une terminaison de la session.

A I.3.2. Contraintes de Performances.

A I.3.2.I. Contraintes de base.

Voir A I 2.2.

A I.3.2.1. Contraintes particulières.

L'utilisateur géographique devra pouvoir facilement coordonner les opérations particulières à l'application géographique.

A I.4 Phase n°2C : Coordination - Version chimique.

A I.4.1. Objectifs.

Dans le cadre de l'application chimique, on veut par cette phase :

- . Assurer la coordination des opérations sur une surface chimique,
 - en permettant l'introduction et le chargement des données des surfaces sur lesquelles l'utilisateur opère,
 - en aiguillant l'utilisateur dans son choix d'opération et de surface chimique visée par l'opération,
 - en assurant un point de retour après terminaison de l'opération.
- . Assurer la fin d'exécution de la phase de coordination dans le but d'une terminaison de la session.

A I. 4.2. Contraintes de Performances.

A I.4.2.1. Contraintes de base.

Voir A I. 2.2.

A I. 4.2.2. Contraintes particulières.

L'utilisateur chimique devra pouvoir facilement coordonner les opérations particulières à l'application chimique.

A I. 5. Phase n°3 : Etude d'une surface.

A I. 5.1. Objectifs

- . Assurer la visualisation et la gestion des différents attributs de visualisation (visualisation en "vu et caché", affichage permanent de points, affichage permanent des axes X, Y et Z, modification de l'orientation des axes, visualisation en bloc-diagramme, visualisation avec isolignes, visualisation des vecteurs unitaires pour une surface chimique, visualisation des atomes de la molécule par une surface chimique).
- . Assurer les manipulations interactives (rotation, translation, coupe, accès à un point, fenêtrage, changement d'échelle) de la surface visualisée.

. Assurer la gestion de l'étude de la surface (mémorisation, impression, retour à la surface avant manipulation).

A I. 5.2. Contraintes de Performances.

A I.5.2.1. Contraintes de base.

Voir A 5.1.2.1.

A I.5.2.2. Contraintes particulières.

. La visualisation des surfaces définies par des points et ayant subi une interpolation locale ou une approximation de Brunet ou encore une approximation par "patches", se fera par "wire-frame". Il en sera de même pour les surfaces définies par des équations. Les surfaces chimiques seront elles visualisées par points, d'intensité variable selon la coordonnée-écran Z_S .

. La visualisation sera la plus rapide possible, la plus claire et la plus centrée.

. Les manipulations seront réalisées rapidement et facilement.

A I.6. Phase n°4 : Etude de deux surfaces.

A I.6.1. Objectifs.

Assurer l'étude comparative de deux surfaces quelconques qui nécessite d' :

. assurer la visualisation et la gestion des différents attributs de visualisation des deux surfaces à l'étude.

. assurer les manipulations interactives des deux surfaces visualisées.

. assurer les comparaisons spatiale (addition ou soustraction) et/ou statistique (écart entre les coordonnées des points des deux surfaces).

. assurer la gestion de l'étude des deux surfaces (mémorisation d'une surface, d'une comparaison spatiale; impression d'une surface, d'une comparaison spatiale; retour à une surface avant manipulation; retour aux surfaces avant comparaison).

A I.6.2. Contraintes de Performances.

A I. 6.2.1. Contraintes de Base.

Voir A I.5.2.1.

A I.6.2.2. Contraintes particulières.

Lors de la visualisation des deux surfaces, on les visualisera à des endroits distincts de l'écran. On pourra facilement manipuler l'une ou l'autre surface au choix.

A I.7. Phase n°5 : Introduction, Consultation et Mise à jour d'une surface.

A I.7.1. Objectifs.

. Assurer la création, la consultation, la modification et la suppression des données constitutives d'une surface (toute version confondue) ainsi que des renseignements relatifs à ces surfaces.

. Assurer la souplesse, au niveau de la création des données constitutives (points ou équation), une facilité d'introduction par l'utilisateur.

. Assurer la validité des quatre opérations citées plus haut, afin d'assurer la cohérence des données et renseignements existant dans le système.

A I.7.2. Contraintes de Performances.

Voir A I.1.2.1.

A I.8. Phase n°6 : Modification de Représentation.

A I.8.1. Objectifs.

. Assurer l'optimisation de la représentation d'une surface existante constituée par un ensemble de points. En Effet, une surface définie ainsi se devra d'avoir une représentation plus efficiente pour sa mémorisation et sa visualisation.

Cette optimisation obligatoire se fera pour les surfaces :

. SIMPLE-VALEUR, par une méthode au choix, d'interpolation locale visant à la construction d'une matrice (méthodes de Beguin(1979) et Jouret(1976), au choix) ou d'approximation globale visant à la construction d'un ensemble de matrices de taille réduite (Brunet (1980))

. MULTIPLE-VALEUR, par la méthode des "parametric patches" visant à la création d'équations paramétriques. Cette méthode est à approfondir pour réaliser son implémentation. Intégrée dans notre logiciel, elle pourrait d'ailleurs permettre une représentation optimale pour tous les types de surfaces définies par des points (Foley, 1980, p.523). Dans une première version, nous ne l'implémenterons pas.

Cette optimisation remplacera la représentation existante selon le choix de l'utilisateur.

. Assurer la création des données et renseignements de la nouvelle représentation d'une surface existante, représentation issue d'une des méthodes décrites ci-dessus.

A I. 8.2. Contraintes de Performances.

Cette phase pourra présenter des traitements plus lents vu leur nature. La contrainte du temps de réponse citée au paragraphe A I.1.2.1. est donc supprimée. Les autres restent.

A I.9 Phase n°7 : Aide à l'utilisateur.

A I.9.1. Objectifs.

L'objectif de cette phase est de permettre à l'utilisateur, l'accès à une documentation. La documentation fournie décrira l'état d'exécution du logiciel au moment de l'appel. Seront fournis les renseignements suivants :

- . le niveau de l'interaction en cours en termes de phases et fonctions où cette interaction se situe,
- . la dernière commande effectuée avant l'appel (HELP),
- . les commandes possibles, avec une explication de celles-ci,
- . les noms des surfaces dont les données sont chargées.

A I.9.2. Contraintes de Performances.

A I.9.2.1. Contraintes de base.

Voir A I.1.2.1.

A I. 9.2.2. Contraintes particulières.

L'appel à la phase d'aide doit répondre à plusieurs exigences :

- . il ne peut y avoir de perte de renseignements nécessaires à l'exécution en cours, du logiciel.
- . il doit y avoir un retour à l'endroit d'appel, après consultation de la documentation.
- . l'appel ne sera possible que lorsque l'utilisateur est en attente dans un dialogue.

A I. 10. Phase n°8 : Terminaison d'une session.

A I.10.1. Objectifs.

- . Assurer la terminaison d'une session de travail
 - en produisant les résultats de celle-ci,
 - en demandant la confirmation de cette terminaison :
 - * si oui, on assurera le lancement des travaux batch, issus de la session, et la fin de connexion du logiciel.
 - * si non, on assurera le retour en session de travail dans une version.
 - . Assurer la fin de connexion en cas de non-validation de l'identification de l'utilisateur.

A I. 10.2. Contraintes de Performances.

Voir A I. 1.2.1.

A I. 11. Phase n°9B : Traitements Complémentaires - Version géographique

A I. 11.1. Objectifs.

Assurer, dans le cadre de la version géographique, les traitements non liés à l'étude d'une surface originale géographique mais nécessaires au vu des objectifs de l'application géographique (voir paragraphe II)

Ces traitements sont principalement :

- la création de surfaces modélisées à partir d'une surface originale géographique.

- le calcul des résidus de cette modélisation par le calcul de l'écart entre une variation réelle de densités et leur variation modélisée.

A I. 11.2 Contraintes de Performances.

Voir A I. 1.2.1.

ANNEXE II :

Les sous - schémas conceptuels des données et des traitements.

A II.0 Remarques préliminaires.

Avant d'entamer la description des sous-schémas conceptuels, faisons quelques remarques préliminaires.

Les sous-schémas conceptuels des données décrivent, par le modèle entité-association, les données permanentes liées à une phase. Dans le cadre de notre projet les données sont souvent utilisées au sein de plusieurs phases. C'est pourquoi des éléments (entité, association, attribut, message) d'un même sous-schéma conceptuel de données pourront servir à plusieurs phases.

En ce qui concerne les messages de notre projet, ceux-ci (au sens Bodart) ne reprennent que des informations véhiculées au sein du projet, contenues dans la B.D. permanente.

En plus de ces messages, nous spécifierons des pseudo-messages (notés "Messages") reprenant des informations non-permanentes (ou variables) nécessaires à l'exécution de fonctions ou de phases de notre projet.

Ces "messages" ne seront précisés que dans la statique des fonctions et non dans les schémas conceptuels. Ceci pour suivre les consignes de la méthodologie prônée par Bodart.

Au sein des sous-schémas conceptuels des traitements, nous détaillerons entre autres, la dynamique des fonctions de la phase. Ce schéma de la dynamique ne comportera que les messages externes, à la phase, c'est-à-dire, issus de l'extérieur de la phase ou allant vers une autre phase. Nous décrivons en effet, cette dynamique par le modèle de la statique, en prenant comme référence la phase. On trouvera dans la description de la statique de fonctions, la spécification des messages entrants et sortants par rapport aux fonctions.

A II.1 Phase n°I : Initialisation d'une session.

A II.1.1 Sous-schéma des données (figure 26)

. Entité UTILISATEUR. Un utilisateur correspond à une personne qui a été autorisée par un utilisateur privilégié à entrer dans le système et à en utiliser une ou plusieurs versions.

. Entité VERSION : Une version correspond à un ensemble de ressources de traitements afin d'atteindre les objectifs fixés à cette version.

. Elément NOMUSER : Un nomuser correspond au nom de l'utilisateur
Format : 12 alphabétiques.

. Elément MOT DE PASSE : Un mot de passe est attaché à un seul utilisateur (identifiant). Il est sans signification. Format : 6 alphanumériques.

. Elément PRIVILEGE : Le privilège indique qu'un utilisateur peut modifier les entités UTILISATEUR. Format : booléen.

. Elément NON-VERS : Le numéro de version identifie (identifiant) la version.
Format : 2 alphanumériques.

. Elément DATE-CREA-VERS : La date de création d'une version correspond à la date de mise en oeuvre complète des ressources de traitement d'une version.
Format : 10 alphanumériques.

. Elément ETAT-VERS : L'état d'une version indique la complétude ou non des ressources d'une version. Format : booléen.

. Elément NON-CREA-VERS : Le nom du créateur de la version est un utilisateur ou non de celle-ci, qui l'a réalisée. Format : 12 alphabétiques.

. Association UTILISE : Cette relation montre l'utilisation d'une version par un utilisateur du logiciel.

. Message IDENTIFICATION : Constitué d'une occurrence de MOT DE PASSE et de NON-VERS, ce message correspond à l'identification d'un utilisateur et de son choix de version.

A II.1.2. Sous-schéma des traitements.

a. Mise en Oeuvre.

Après l'appel à l'exécution du logiciel, on procédera à l'introduction du nom de l'utilisateur et de son mot de passe et de la version choisie. La validation de ses introductions amènera au test des ressources de traitement de la version choisie et aux ouvertures de fichiers et initialisations nécessaires. Ensuite, l'utilisateur sera placé en phase de coordination.

La non-validation des introductions d'identification conduira l'utilisateur à sortir du système par le biais de la phase 8.

Si l'utilisateur est privilégié, il pourra avant de passer en coordination, mettre à jour les caractéristiques des utilisateurs de sa version et les caractéristiques de celle-ci (état des ressources notamment).

b. Découpe en fonctions (Voir tableau 2).

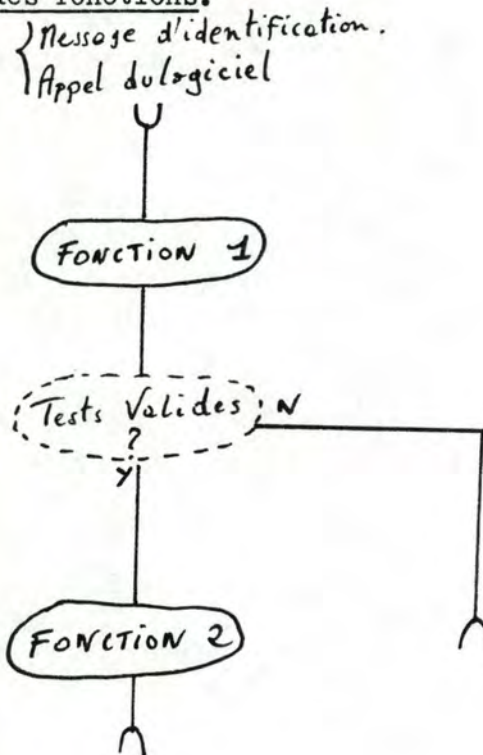
On peut facilement identifier deux fonctions. Celle regroupant les initialisations interactives et les tests qui y sont liés.

Ensuite une fonction réalisant les traitements d'initialisations non-interactives.

La première pourrait se résumer sous l'expression "identifier l'utilisateur et sa session".

La seconde fonction elle, correspondrait à "initialiser la version du système".

c. Dynamique des fonctions.



d. Statique des fonctions.

* FONCTION 1 : Introduction et tests des caractéristiques d'exécution.

- . condition d'activation : appel à l'exécution du logiciel.
- . message en entrée : message d'indentification.
- . règles de traitement : Le message d'indentification sera testé en consultant la liste des utilisateurs.

1 Si celle-ci contient l'utilisateur identifié, alors on teste s'il est associé avec la version choisie :

- . Si oui, on teste la validité des ressources de celle-ci en consultant la liste des versions et de leur état.
- . Si non, on refuse l'entrée dans la version, on demande un nouveau message d'identification, et des explications du blocage sont fournies.

Si les ressources sont valides, on permet l'entrée dans la version par un "message" positif d'entrée dans la version.

Si les ressources ne le sont pas, on refuse l'entrée; on demande un nouveau message d'indentification.

2 Si l'identification n'est pas correcte, après trois essais d'introduction, on refuse l'entrée dans la version par la production d'un "message" négatif d'entrée dans la version.

3 Si l'utilisateur identifié est validé (tests d'existence et de ressources positifs) et privilégié alors on traitera un message de gestion des utilisateurs si l'utilisateur le désire; et ceci avant l'entrée dans la version. Ce message, après validation, permettra la lecture et la mise à jour des listes d'utilisateurs et de versions.

. message en sortie : "message" d'entrée dans la version, positif (avec num-vers = $[0,2]$) ou négatif (avec num-vers = $[0,2]$ et indice de cause d'échec (identification mauvaise, ressources invalides, autres))

* FONCTION 2 : Initialisations du système.

. condition d'activation : "message" d'entrée dans la version dont le num-vers = $[0,2]$.

. message en entrée : "message" positif d'entrée dans la version.

. règles de traitement: - "message" d'entrée permise induira l'ouverture des fichiers permanents contenant les données et les renseignements des surfaces de la version.

- on produira un "message" d'appel à la phase 2A, 2B ou 2C selon le num-vers.

. message en sortie : "message" d'appel à la phase 2 dont le num-vers = $[0,2]$.

A II.2. Phase n° 2A : Coordination - Version Simple.

A II.2.1. Sous-schéma des données (figure 26)

. Entité SURFACE-ORIGINALE : Une surface originale est une surface à laquelle sera associée un ensemble de points ou d'équations et qui est issue, sans une quelconque modification de représentation, de l'introduction par l'utilisateur des données. Cette surface originale est définie, dans le cadre de la version simple, par une occurrence de l'entité SURFACE-ORIGINALE et par des occurrences de l'entité POINT (si le type de définition = 'P') ou de l'entité EQUATION (si le type de définition = 'E').

. Elément NOM.O : Un nom-o correspond simplement au nom de la surface Originale. Format: 6 alphanumériques.o

. Elément TYPE-DEFINITION: Un type de définition correspond au type de données qui définiront la surface originale : des points ou des équations. Format: 1 alphabétique.

. Elément SENS-AXES : L'attribut "sens axes" précise dans quel système d'axes, left-handed ('L') ou right-handed ('R'), la surface est à visualiser par défaut. Format: 1 alphabétique.

. Elément TYPE-SURFACE : Un type de surface correspond au caractère simple-valeur ('SV') ou multiple valeur ('MV') de la surface originale. Format: 2 alphabétiques.

. Elément NOM-CREA-SURF : L'attribut "nom-crea-surf" reprend le nom de la personne qui a introduit la surface dans le système. Format: 12 alphabétiques.

. Entité POINT : L'entité point reprend les éléments définitionnels d'un point d'une surface. Plusieurs mesures pourront être réalisées en chaque point, un point pourra se voir attribuer, plusieurs coordonnées.

. Elément COORDONNEE : L'attribut coordonnée reprend les trois valeurs d'une coordonnée. Format : 3 (6 numériques - 2 numériques)

. Entité EQUATION : L'entité équation reprend les éléments définitionnels d'une équation d'une surface.

. Elément TYPE-EQUATION : Cet élément précise le degré de l'équation cartésienne (1er et 2ème). Format: 2 alphanumériques.

. Elément COEFFICIENTS : Cet élément reprendra les coefficients (4 ou 10 selon le degré) de l'équation. Format : variable 4 ou 10 (6 numériques).

. Entité SURFACE-ORIGINALE-APPROXIMEE : Une surface originale approximée est une surface originale de type de définition 'P', mais qui, par la méthode de Brunet, a subi une approximation globale menant à un ensemble de matrices.

Cet ensemble de matrices pourra être représenté par un ensemble de points. Les trois valeurs d'une coordonnée correspondront alors aux indices ligne et colonne et à la valeur d'un élément de la matrice.

Cette surface originale approximée est définie par une occurrence de l'entité SURFACE-ORIGINALE-APPROXIMEE et par des occurrences de l'entité POINT.

. Elément NOM.O.A.NO. : Cet attribut désigne le nom de la surface originale approximée. La partie d'élément NOM est identique au nom de la surface originale "mère". La partie d'élément N° (≥ 1) indique le numéro de l'occurrence de surface originale approximée issue de la même surface originale "mère" par le biais de différentes approximations. Format : 6 alphanumériques O.A. 2 numériques.

. Elément METHODE-A : Cet attribut reprendra les renseignements essentiels de la mise en oeuvre de la méthode de Brunet. Format : 64 alphanumériques.

. Entité SURFACE-ORIGINALE-MATRICIELLE : Une surface originale matricielle est une surface originale de type de définition 'P', mais qui, par la méthode de Beguin ou de Jouret, a subi une interpolation locale menant à une matrice.

Cette matrice sera également représentée par un ensemble de points.

Une surface originale matricielle sera définie par une occurrence de l'entité SURFACE-ORIGINALE-MATRICIELLE et par des occurrences de l'entité POINT

. Elément NOM.O.M. N° : Cet attribut désigne le nom de la surface originale. La partie d'élément NOM est identique au nom de la surface originale "mère". La partie d'élément n° (≥ 1) indique le numéro de l'occurrence de surface originale matricielle issue de la même surface originale "mère" par le biais de différentes interpolations. Format : 6 alphanumériques. O.M. 2 numériques.

. Elément METHODE - I : Cet attribut reprendra les renseignements essentiels de la mise en oeuvre de la méthode de Beguin ou de Jouret. Format : 64 alphanumériques.

. Elément MAILLE : Cet attribut désignera la taille de la maille d'interpolation, mesurée dans les unités de mesures des coordonnées.

. Entité MANIPULATION : Cette entité identifie la notion de manipulation d'une surface. La surface manipulée pourra être :

- * une surface originale quelconque, ou deux surfaces originales s'il y a eu comparaison spatiale.
- * une manipulation quelconque, ou deux manipulations s'il y a eu comparaison spatiale.
- * une surface originale et une manipulation s'il y a eu comparaison spatiale.

Une manipulation est définie par une occurrence de l'entité MANIPULATION.

Remarques importantes : - Nous n'avons pas créé l'entité "surface manipulée" au lieu de "manipulation" car on n'associe pas de données de définition à la manipulation. D'où lors, de la mémorisation d'une manipulation, on ne garde que les paramètres de celle-ci et non les données modifiées de la surface "mère".

- Dans le cas où l'utilisateur manipule une manipulation (1) (et crée donc une manipulation (2)), la mémorisation de (2) prendra pour surface origine la surface originale mère de (1) et non la manipulation (1). Il y a transivité dans l'origine de la manipulation (2); C'est pourquoi, il nous a semblé inutile de créer une association récurrente du type "ORIGINE-DE" sur l'entité MANIPULATION.

- . Elément NOM.M.NO : Cet attribut désigne le nom de la manipulation.

La partie d'élément NOM sera quelconque. La partie d'élément NO (≥ 1) indique le numéro de l'occurrence de manipulation issue de la même surface "mère".

- . Elément TYPE-MANIP : Cet attribut reprend la liste des manipulations effectuées sur la ou les (dans le cas d'une comparaison), surface(s) origine(s).

- . Elément SURF-ORIGINE : Cet attribut identifie les noms de la ou les surface(s) "mère(s)" de la manipulation.

- . Association DEFINIE-PAR-1 : Cette relation traduit le lien entre une surface originale de type de définition 'E' et les équations qui la définissent.

- . Association DEFINIE-PAR-2 : Cette relation traduit le lien entre une surface originale de type de définition 'P' et les points qui la définissent.

- . Association DEFINIE-PAR-3 : Cette relation traduit le lien entre une surface originale approximée et les points qui la définissent.

- . Association DEFINIE-PAR-4 : Cette relation traduit le lien entre une surface originale matricielle et les points qui la définissent.

- . Association ORIGINE-DE-1 : Cette relation traduit la liaison de "maternité" entre une surface originale et sa modification de représentation par le biais d'une approximation globale.

. Association ORIGINE-DE-2 : Cette relation traduit la liaison de "Maternité" entre une surface originale et sa modification de représentation par le biais d'une interpolation locale.

. Association ORIGINE-DE-3 : Cette relation traduit la liaison de "Maternité" entre une surface originale et une manipulation de celle-ci.

. Association ATTACHEE-A-1 : Cette relation traduit qu'une surface originale est définie dans le cadre d'une version. Dans la phase 2A, c'est la version simple.

. Message APPEL-PH-3 : Ce message contiendra les renseignements généraux et les données originales de la surface à étudier.

. Message APPEL-PH-4 : Ce message sera porteur des renseignements et données des surfaces à étudier.

. Message APPEL-PH-5 : Ce message sera porteur des renseignements et données de la surface à mettre à jour.

. Message APPEL-PH-6 : Ce message sera identique à celui de l'appel à la phase 3.

. Contraintes d'intégrité sur le sous-schéma :

* Si la surface originale est de type définition 'P', elle ne pourra être définie que par un ensemble de points non par des équations (et vice et versa).

* Les surfaces originales approximée et matricielle ne sont associées qu'avec des surfaces originales de type de définition 'P'.

A II.2.2. Sous-schéma des traitements.

a. Mise en oeuvre.

Venant de la phase d'initialisation réalisée, on choisira tout d'abord si l'on désire opérer sur une ou deux surfaces.

- . Si une surface : On choisira ensuite si l'on veut
 - étudier une manipulation (1)
 - opérer sur une surface originale (2)
- Si (1) : On préparera alors l'appel à la phase 3,
 - en indiquant que l'on étudie une surface manipulée
 - en introduisant un choix valide de manipulation après
 - en avoir consulté les caractéristiques (renseignements).

en chargeant les données de la surface-mère originale associée.

On appellera enfin la phase 3.

- Si (2): On préparera l'appel à une phase suivante

en choisissant entre une étude d'une surface originale

une introduction d'une surface originale

une consultation et mise à jour d'une

surface originale.

une modification de représentation d'une

surface originale

en effectuant, hormis dans le cadre d'une introduction,

* l'introduction d'un choix valide de surface originale
après en avoir consulté les renseignements.

* le chargement des données de la surface originale

* si cette surface est du type 'P' alors on fera un appel
obligatoire à la phase 6 pour modifier la représentation,
pour autant qu'il n'existe pas encore de surface approximée
ou matricielle associée à la surface originale.

On appellera, suivant le choix effectué :

la phase 5 : introduction d'une surface originale,

la phase 5 : consultation et mise à jour d'une
surface,

la phase 3 : étude d'une surface originale,

la phase 6 : modification de la représentation
d'une surface originale.

- Si deux surfaces : On choisira ensuite si l'on veut

- étudier deux manipulations (1)

- étudier une manipulation et une surface originale(2)

- étudier deux surfaces originales.(3)

Dans les trois cas, on préparera l'appel à la phase 4,
en indiquant le type d'étude.

en introduisant un choix valide de deux surfaces après en
avoir consulté les renseignements

en chargeant les données concernées des surfaces originales
concernées

Si une des deux surfaces est du type 'P' alors on fera appel à la phase 6 pour autant qu'il n'existe pas encore de surface approximée ou matricielle associée à la surface originale.

On appellera ensuite la phase 4.

En l'absence de choix d'opération sur une ou deux surfaces, l'utilisateur quittera (avec un dispositif de minuterie) le coordinateur pour terminer la session de travail (appel à la phase 8).

b. Découpe en fonctions. (Voir tableau 2).

Selon la mise en oeuvre de la phase, on peut reconnaître trois fonctions.

Ces trois fonctions correspondent à trois actions élémentaires et distinctes pour l'utilisateur dans la coordination:

- " Choisir la ou les surfaces à mettre en opération" (1)
- " Charger en mémoire central les données (Points ou équations) associées aux surfaces originales en opération "(2)
- " Choisir le type d'opération sur une surface originale" (3)

c. Dynamique des fonctions

. Voir page n°125.

d. Statique des fonctions.

* Fonction 1 : Choix des surfaces à opérer.

. condition d'activation: existence d'un "message" d'appel à la phase 2A où le num-vers = ϕ , ou d'un "message" d'appel à la fonction 1.

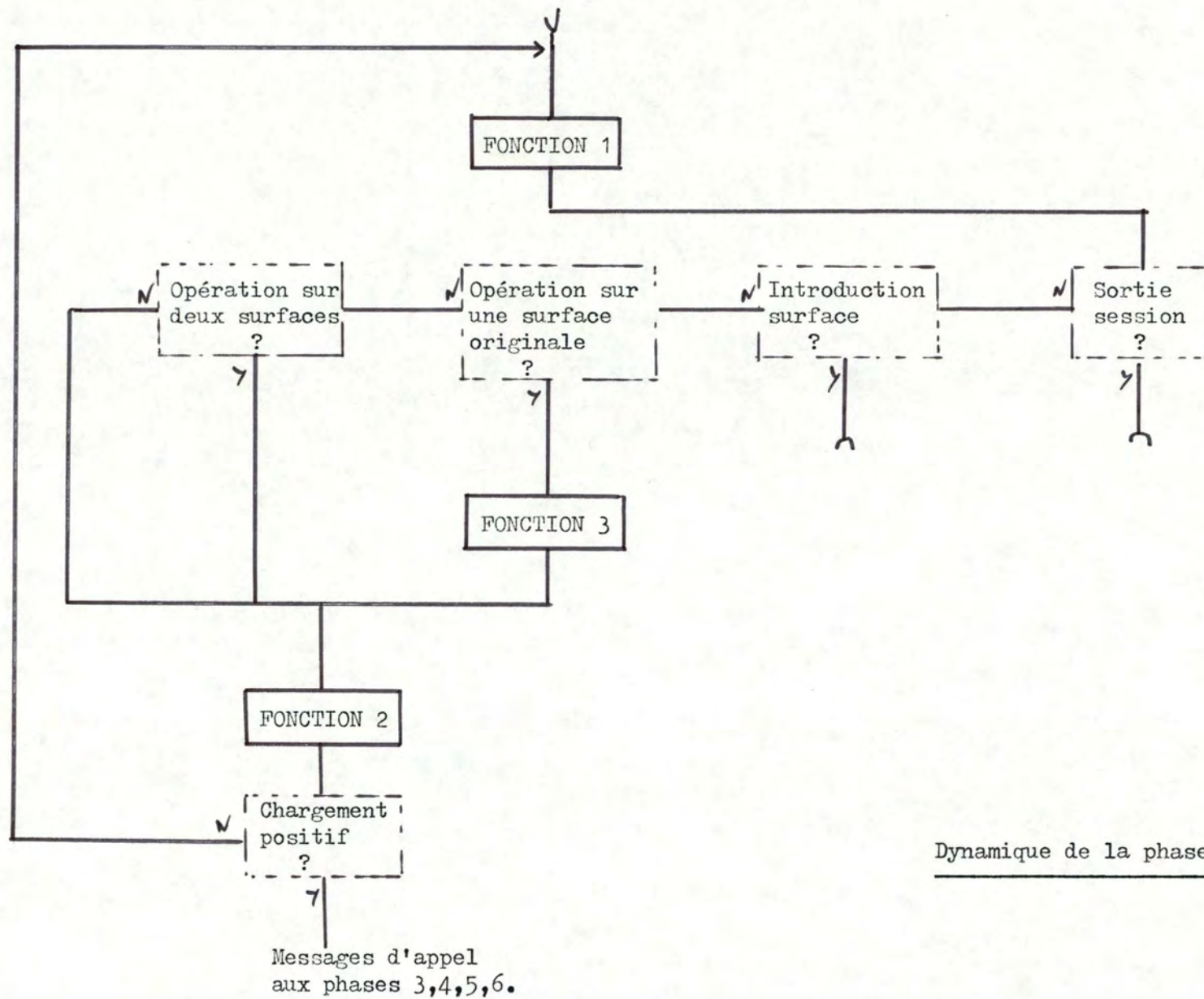
. message en entrée : "message " d'appel à la phase 2A, ou message d'appel à la fonction 1.

. règles de traitements: - on va lire et afficher la liste des noms des surfaces de la version simple (élément NOM.O de l'entité SURFACE-ORIGINALE ; élément NOM.M.NO. de l'entité MANIPULATION ; élément NOM.O.A.NO. de l'entité SURFACE-ORIGINALE-APPROXIMÉE; élément NOM.O.M.NO. de l'entité SURFACE-ORIGINALE-MATRICIELLE.)

- on va choisir une des options suivantes:

- . Sortie de la session (1),
- . Introduction d'une nouvelle surface(2);

OU, on va choisir un ou deux noms de surfaces



Dynamique de la phase 2A.

(une surface originale (3), une manipulation (4), deux surfaces originales (5), deux manipulations (6), une surface originale et une manipulation (7)). Donc lors de ce choix de noms de surfaces, l'utilisateur pourra consulter les renseignements concernant la surface (Voir spécification du module n°63 en Annexe III). Au sein des possibilités n° (5) et (7), une surface originale de type de définition 'P' ne pourra être choisie sans avoir subi une modification de représentation. Dans ce cas, l'utilisateur sera amené à choisir l'option N°(3).

- Après choix d'une des possibilités (n°1 à 7), l'utilisateur validera ce choix et il y aura création de messages d'appel.

- . Messages en sortie: . Si(1): "Message" d'appel à la phase 8.
- . Si(2): "Message" d'appel à la phase 5 pour introduire une surface.
- . Si(3): "Message" d'appel à la fonction 3.
- . Si(4): "Message" d'appel à la phase 3,
"Message" d'appel à la fonction 2 (type - surface = 'M')
- . Si(5): "Message" d'appel à la phase 4,
Un "Message" d'appel à la fonction 2
(type - surfaces = 'O.O').
- . Si(6): "Message" d'appel à la phase 4.
Un "Message" d'appel à la fonction 2
(type - surfaces = 'M.M')
- . Si(7): "Message" d'appel à la phase 4
Un Message d'appel à la fonction 2
(type - surfaces = 'O.M').

* FONCTION 2 : Chargement des données de la surface originale.

- . condition d'activation : existence d'un "message" d'appel à la fonction 2.
- . message en entrée : "Message" d'appel à la fonction 2 et "message" d'appel aux phases (3,4,5,6).
- . règles de traitement: - on va charger en Mémoire centrale les données des surfaces originales associées aux surfaces choisies dans la fonction 1.

. Si type-surface = 'O' alors on chargera les données associées au nom.O.* du "message" d'appel (nom.O.* = [nom-O] ou [nom.O.A.NO] ou [nom.O.M.NO.]).

. Si type-surface = 'M' alors on chargera les données associées aux noms issus de l'élément SURF-ORIGINE, de la manipulation nom.M.NO. du "message" d'appel.

. Si type-surfaces = 'O.O' alors on chargera les données associées aux noms O.* du "message" d'appel.

. Si type-surfaces = 'M.M' alors on chargera les données associées aux noms issus de l'élément SURF-ORIGINE, des manipulations noms.M.NO. du "message" d'appel.

. Si type-surface = 'D.M' alors on chargera d'une part les données associées au nom.O.* du "message" d'appel, et d'autre part, les données associées aux noms issus de l'élément SURF-ORIGINE, de la manipulation nom.M.NO. du "message" d'appel.

Remarque: Voir module n°64 en annexe III, pour le détail de la notion de "données originales".

- on informera l'utilisateur de la validité du chargement
- si le chargement n'est pas valide, alors on engendrera un "message" d'erreur et un "message" d'appel à la fonction 1. Sinon, les messages (au sens propre) d'appel aux phases sont produits.

. Message en sortie : Message d'appel aux phases (3,4,5,6).

OU

"Message" d'appel à la fonction 1.

* FONCTION 3 : Initialisations de l'opération sur une surface originale.

. condition d'activation : existence d'un "message" d'appel à la fonction 3.

. message en entrée : "message d'appel à la fonction 3.

. règles de traitement : - on choisira, après affichage du nom.O.* du "message" d'appel, entre :

- . étude de la surface (1)
- . mise à jour et consultation des données et renseignements de la surface (2)
- . modification de la représentation (3)

- Si nom.O.* = nom.O. et que la surface est de type de définition = 'P', alors la modification de représentation sera obligatoire. Dans une première mise en oeuvre du logiciel seule les surfaces du type 'SV', pourront subir cette modification obligatoire.

- On générera les messages d'appel.

- . Messages en sortie : . Si(1) : "Message" d'appel à la phase 3'
- . Si(2) : "Message" d'appel à la phase 5 pour mettre à jour une surface
- . Si(3) : "Message" d'appel à la phase 6.
- ET, "Message" d'appel à la fonction 2 (type surface = 'O').

A II.3 Phase 2B : Coordination - Version géographique.

A II.3.I. Sous-schéma des données (figure 26).

Les données permanentes utilisées dans cette phase sont proches de celles utilisées au cours de la phase 2A (Voir A II.2.1.).

Deux différences majeures sont à noter.

La première est issue du caractère géographique des surfaces sur lesquelles on opère au sein de la version géographique. Ce caractère implique une contrainte d'intégrité supplémentaire à celles décrites en A II.2.1. :

* les surfaces originales de la version géographique, dites surfaces originales géographiques, ne pourront être que du type de définition 'P' et de type-surface 'SV'.

De plus, les surfaces originales géographiques seront liées à une application géographique spécifiant les caractéristiques générales du phénomène géographique mesuré. D'où, la création de l'entité APPLICATION-GEOGRAPHIQUE.

. Entité APPLICATION-GEOGRAPHIQUE : Cette entité décrit le phénomène géographique associé à la surface originale. Ce phénomène correspond à un site urbain sur lequel on a mesuré la variation des densités de population.

. Élément NOM.GEO. : un nom-geo correspond à la dénomination de l'application associée à la surface du même nom (nom.O). Format : 6 alphanumérique .GEO.

. Elément LIEU : Cet élément désigne le lieu du site urbain. Format : 12 alphabétiques.

. Elément TYPE-CENTRE : Cet attribut indique la méthode de calcul du centre (introduction interactive ou calcul par la méthode de Bussi re et Stovall (1981,P.103). Format : 2 alphab tiques.

. El ment COORD-CENTRE : Cet attribut reprend la valeur de la coordonn e du centre. Format: (6 num riques. 2 num riques).

. El ment TYPE-DONNEES : Cet attribut indique le type de mesures effectu es en chaque point de la surface ('P': mesure des populations; 'D' : mesure des densit s ; 'P.D' : mesures des populations et densit s). Format : 2 alphab tiques.

. Association ASSOCIE - A-1 : Cette relation signifie l'association d'une application g ographique avec une surface mesurant le ph nom ne g ographique.

On peut encore signifier les particularit s suivantes :

* Une application g ographique ne peut  tre associ e qu'  une surface originale g ographique.

* Une surface originale g ographique ne peut  tre associ e qu'  une occurrence de l'entit  VERSION dont l' l ment NUMERO = 1.

La seconde diff rence majeure est l'apparition des surfaces originales mod lis es g ographiques.

. Entit  SURFACE-ORIGINALE-MODELISEE-GEOGRAPHIQUE : Une surface originale mod lis e g ographique est une surface de type de d finition 'E', issue de la mod lisation d'une surface originale g ographique. Elle sera d finie, seulement dans le cadre de la version g ographique, par une occurrence de l'entit  EQUATION et une occurrence de l'entit  SURFACE-ORIGINALE-MODELISEE-GEOGRAPHIQUE.

. El ment NOM.M.GEO.NO. : Cet attribut d signe simplement le nom de la surface g ographique mod lis e.

La partie d' l ment NOM est identique au nom de la surface qui a servi de base   la mod lisation.

La partie d' l ment NO indique la possibilit  de r aliser plusieurs mod lisations   partir de la m me surface. Format : 6 alphanum riques. M.GEO.NO.

. Elément TYPE-MODELE : L'élément identifie le modèle appliqué pour créer la surface (quadratique, newtonien et directionnel). Format : 4 alphabétiques. On y renseignera aussi le centre de calcul du modèle.

. Elément TYPE-MODELISATION : L'élément indique si la modélisation a été réalisée à partir des populations ('P') ou des densités ('O'). Format : 1 alphabétique.

. Elément RESULTATS-MOD : L'attribut reprendra les résultats statistiques au choix de la modélisation. Format : 64 alphanumériques.

. Association DEFINIE-PAR-5 : La relation montre que la surface modélisée fait l'objet de la mémorisation de son équation.

. Association ORIGINE-DE-4 : Cette relation montre le lien de "maternité" entre une surface modélisée et la variation réelle exprimée par la surface originale.

. Association ORIGINE-DE-6 : Cette relation montre la possibilité, puisqu'originale, de manipuler une surface modélisée.

. Message APPEL-PH-9B : Ce message contiendra les renseignements et données de la surface originale pour une modélisation.

Pour un calcul des résidus, il contiendra en plus, les renseignements et données de la surface modélisée associée.

Ajoutons enfin la contrainte d'intégrité suivante :

* Une surface modélisée ne peut être qu'associée avec une surface originale géographique.

AII.3.2. Sous-schéma des traitements.

a. Mise en oeuvre.

Venant de la phase d'initialisation réalisée et où la version géographique fut choisie, on informera tout d'abord si l'on désire opérer sur une ou deux surfaces.

- . Si une surface : On choisira ensuite si l'on veut
 - étudier une manipulation géographique (1)
 - opérer sur une surface originale géographique ou sur une surface originale modélisée géographique (2)
- Si (1) : On préparera alors l'appel à la phase 3
 - en positionnant une variable indiquant que l'on étudiera une manipulation géographique.

en introduisant un choix valide de manipulation géographique après en avoir consulté les renseignements.

en chargeant les données de la surface-mère originale géographique associée.

On appellera enfin la phase 3.

- Si(2) : On préparera l'appel à une phase suivante en choisissant entre une étude d'une surface originale géographique.

une étude d'une surface originale modélisée géographique avec ou sans résidus.

une mise en oeuvre des traitements complémentaires.

une introduction d'une surface originale géographique.

une consultation ou mise à jour d'une surface originale géographique.

une modification de la représentation d'une surface originale géographique

en effectuant, hormis dans le cadre d'une introduction,

* l'entrée du choix valide de surface géographique appropriée après en avoir consulté les renseignements.

* le chargement des données de la surface originale géographique.

* Si cette surface géographique originale n'a pas encore fait l'objet d'une approximation ou interpolation, on fera appel à la phase 6.

On appellera suivant le choix effectué

la phase 5 : Introduction d'une surface originale géographique

la phase 5 : Consultation et mise à jour d'une surface géographique

la phase 6 : Modification de la représentation d'une surface originale géographique.

la phase 9B : Réalisation des traitements complémentaires.

la fonction d'initialisation d'une étude de surface originale modélisée géographique.

la phase 3 : étude d'une surface originale géographique

- Si deux surfaces : On choisira ensuite si l'on veut

- étudier deux manipulations géographiques (1)

- étudier une manipulation géographique et une surface originale géographique.(2).

- étudier deux surfaces originales géographiques(3)

Dans les trois cas, on préparera l'appel à la phase 4

en indiquant le type d'étude.

en introduisant un choix valide de deux surfaces géographiques après en avoir consulté les renseignements de celles-ci.

en chargeant les données des surfaces originales géographiques concernées.

Si une des deux surfaces originales géographiques n'a pas encore fait l'objet d'une approximation ou interpolation, on fera appel à la phase 6.

On appellera ensuite la phase 4.

En l'absence de choix d'opération sur une ou deux surfaces géographiques, l'utilisateur quittera (avec un dispositif de minuterie) le coordinateur pour terminer la session de travail (appel à la phase 8).

b. Découpe en fonctions. (Voir tableau 2).

Selon la mise en oeuvre de la phase, on peut reconnaître quatre fonctions.

Ces quatre fonctions correspondent à quatre actions élémentaires et distinctes pour l'utilisateur, dans la coordination :

- . "choisir la ou les surfaces à mettre en opération" (1).
- . "charger en mémoire centrale les données associées aux surfaces originales en opération" (2).
- . "choisir le type d'opération sur une surface originale géographique" (3).
- . "choisir le type d'opération sur une surface originale modélisée géographique" (4).

c. Dynamique des fonctions.

Voir page n°133.

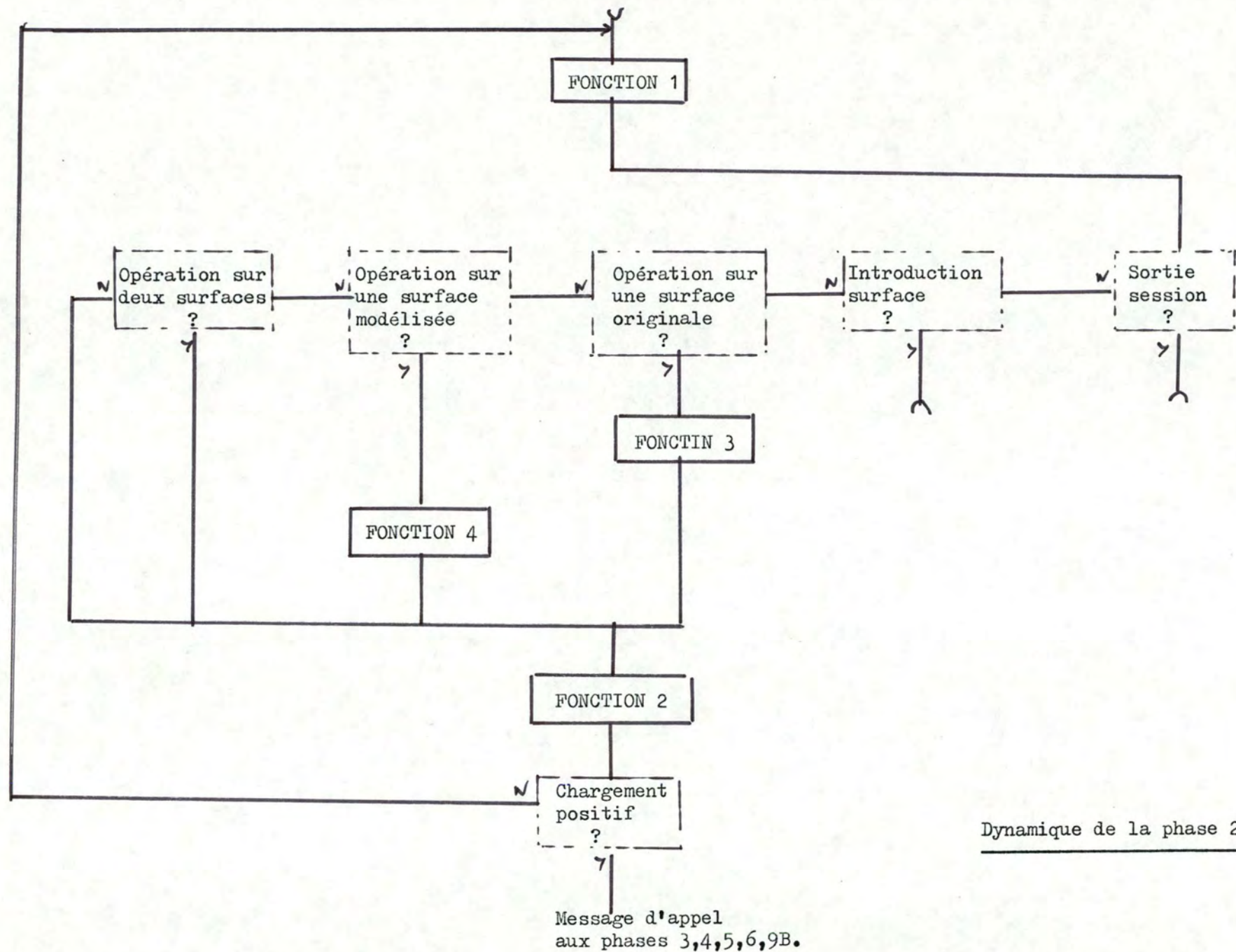
d. Statique des fonctions.

* Fonction 1 : Choix des surfaces géographiques à opérer.

. condition d'activation : existence d'un "message d'appel à la phase 2B, où le num-vers = 1; ou "message" d'appel à la fonction 1.

. message en entrée : "message" d'appel à la phase 2B, ou "message" d'appel à la fonction 1.

. règles de traitement : - on va lire et afficher la liste des noms des surfaces de la version géographique (éléments identiques à la version, simple, en plus de l'élément NOM.M.6GEO.NO. de l'entité SURFACE-ORIGINALE-MODELISEE-GEOGRAPHIQUE).



- on va choisir une des options suivantes :

- . Sortie de la session (1),
- . Introduction d'une nouvelle surface originale géographique (2),

OU, on va choisir un ou deux noms de surfaces (une surface originale (3), une manipulation (4), deux surfaces originales (5), une surface originale et une manipulation (7), deux manipulations (6), une surface modélisée (8), deux surfaces modélisées (5), une surface modélisée et une surface originale (10), une surface modélisée et une manipulation (11)).

Lors de ce choix de noms de surfaces, l'utilisateur pourra consulter les renseignements concernant la surface (Voir spécification du module n°65 en Annexe III).

Au sein des possibilités n° (5) et (7), une surface originale ne pourra être choisie sans avoir subi une modification de représentation. Dans ce cas, l'utilisateur sera amené à choisir l'option n° (3).
- Après le choix d'une des possibilités (n° 1 à 11), l'utilisateur validera ce choix et il y aura création de messages d'appel.

- . Messages en sorties : . Si (1) : "Message" d'appel à la phase 8.
- . Si (2) : "Message" d'appel à la phase 5 pour introduction d'une surface.
- . Si (3) : "Message" d'appel à la fonction 3.
- . Si (4) : "Message" d'appel à la phase 3 et "Message" d'appel à la fonction 2 (type-surface = 'M').
- . Si (5) : "Message" d'appel à la phase 4 et "Message" d'appel à la fonction 2 (type-surface = 'O.O').
- . Si (6) : "Message" d'appel à la phase 4 et "Message" d'appel à la fonction 2 (type - surface = 'M.M')
- . Si (7) : "Message" d'appel à la phase 4 et "Message" d'appel à la fonction 2 (type - surface = 'O.M').

- . Si (8) : "Message" d'appel à la fonction 4.
- . Si (9) : "Message" d'appel à la phase 4 et
"Message" d'appel à la fonction 2
(type- surface = 'M.GEO. M.GEO').
- . Si (10) : "Message d'appel à la phase 4 et
"Message d'appel à la fonction 2
(type - surface = 'M.GEO. 0')
- . Si (11) : "Message d'appel à la phase 4 et
"Message" d'appel à la fonction 2
(type - surface = 'M.GEO.M).

* Fonction 2 : Chargement des données de la surface originale géographique.

Voir fonction 2 de la phase 2A (AII.2.2.d). On devra simplement préciser que le message d'appel peut contenir des types de surfaces supplémentaires ('M.GEO. M.GEO', 'M.GEO.0', 'M.GEO.M', 'M.GEO') et correspondant aux chargements des données originales différentes (spécifiées au module n°64 en annexe III).

Les messages d'appel aux phases (en sortie) pourront s'adresser aux phases n°3,4,5,6 et 9B.

* Fonction 3 : Initialisation sur une surface originale géographique non-modélisée.

. condition d'activation : existence d'un "message" d'appel à la fonction 3.

. message en entrée : "message" d'appel à la fonction 3.

. règles de traitement : - on choisira, après affichage du nom.0.* du message d'appel, entre :

- . étude de la surface (1)
- . mise à jour et consultation des données et renseignements de la surface (2).
- . modification de la représentation (3).
- . modélisation interactive (4)
- Si nom.0.* = nom.0, alors la modification est obligatoire.

- on générera les messages d'appel.

. Messages en sortie : Si (1) : "Message" d'appel à la phase 3.

Si (2) : "Message" d'appel à la phase 5 pour mise à jour.

- . Si (3) : "Message" d'appel à la phase 6.
- . Si (4) : "Message" d'appel à la phase 9B pour modélisation interactive.
- . " Message" d'appel à la fonction 2 (type surface='0')

* Fonction 4 : Initialisation de l'opération sur une surface originale géographique modélisée.

- . condition d'activation : existence d'un "message" d'appel à la fonction 4.
- . message d'entrée : " message " d'appel à la fonction 4.
- . règles de traitement:→ on choisira, après affichage du nom.M.GEO.NO. du message d'appel, entre:

- . étude avec résidus.(1).
- . étude sans résidus.(2).
- . mise à jour et consultation des données et renseignements de la surface (3).
- on générera les messages d'appel.

. Message en sortie : . Si (1) : "Message" d'appel à la phase 9B. pour calcul des résidus.

- . Si (2) : "Message" d'appel à la phase 3.
- . Si (3) : "Message" d'appel à la phase 5 pour mise à jour.
- . "Message" d'appel à la fonction 2 (type-surface 'M.GEO').

A II.4. Phase 2C : Coordination - Version Chimique.

A II.4.1. Sous-schéma des données (figure 26).

Les données permanentes utilisées dans cette phase sont plus restreintes que celles utilisées dans la phase 2A.

Seules les entités SURFACE-ORIGINALE et MANIPULATION ainsi que les associations attenantes sont nécessaires pour décrire les surfaces originales chimiques et les manipulations chimiques. Ceci est dû à la non-implémentation de la phase de modification de représentation.

Parallèlement à la version géographique, le caractère chimique des surfaces sur lesquelles on opère au sein de la version chimique, implique une contrainte d'intégrité particulière :

* Les surfaces originales de la version chimique, dites surfaces originales chimiques, ne pourront être^{que} du type de définition 'P' et de type - surface 'MV'.

Les surfaces originales chimiques seront liées à une application chimique spécifiant les caractéristiques générales du phénomène chimique mesuré.

D'où la création de l'entité APPLICATION-CHIMIQUE..

. Entité APPLICATION-CHIMIQUE : Cette entité décrit le phénomène chimique associé à la surface originale. Ce phénomène correspond à une molécule dont on a mesuré un ensemble de points de sa surface par l'algorithme de Connolly (1982).

. Elément NOM. CHIM. : Cet élément désigne l'application chimique. La partie d'élément NOM sera identique au nom de la surface originale. Format: 6 alphanumériques. CHIM.

. Elément NBRE-ATOMES : Cet attribut comptabilise le nombre d'atomes au sein de la molécule. Format 6 Numériques.

. Elément DENSITE - REQUISE : Cet attribut indique la densité de calcul de points de la surface souhaitée. Format : 3 numériques. 2 numériques.

. Elément NBRE-TYPES-ATOMES : Cet attribut indique le nombre de types atomes présents dans la molécule. Format : 3 numériques.

. Elément RAYON-SONDAGE : Cet attribut indique le rayon de sondage utilisé dans le calcul de Connolly. Format : 3 numériques. 2 numériques.

. Association ASSOCIE - A-2 : Cette relation traduit la liaison entre la surface qui mesure un phénomène décrit par l'application.

AII.4.2. Sous-schéma des traitements

a. Mise en oeuvre.

Cette mise en oeuvre est proche de celle de la phase 2A (Voir AII.2.2.a.)

Trois différences sont à noter :

- . On travaillera avec des surfaces originales chimiques et des manipulations chimiques.
- . On n'implémentera pas la modification de représentation (Phase 6)
- . On ne pourra pas mettre à jour, ni consulter les données de la surface (Phase 5, mise à jour).

b. Découpe des fonctions (Voir tableau 2):

Selon la mise en oeuvre de la phase, on peut reconnaître trois fonctions.

Ces trois fonctions correspondent à trois actions élémentaires et distinctes pour l'utilisateur, dans la coordination :

- . " Choisir la ou les surfaces à mettre en opération["](1).
- . " Charger en mémoire centrale les données (Points) associés aux surfaces originales en opération "(2).
- . " Choisir le type d'opération sur une surface originale " (3).

c. Dynamique des fonctions.

Voir page n° 139.

d. Statique des Fonctions.

* Fonction 1 : Choix des surfaces chimiques à opérer.

Elle est identique à la fonction 1 de la phase 2A, si ce n'est :

- . que le num - version du message d'appel à la phase 2c = 2.
- . que la liste des noms des surfaces de la version chimique ne comprend pas des surfaces nom.O.A.NO. et nom.O.M.NO.
- . qu'une surface originale ne devra pas faire l'objet d'une modification de représentation.

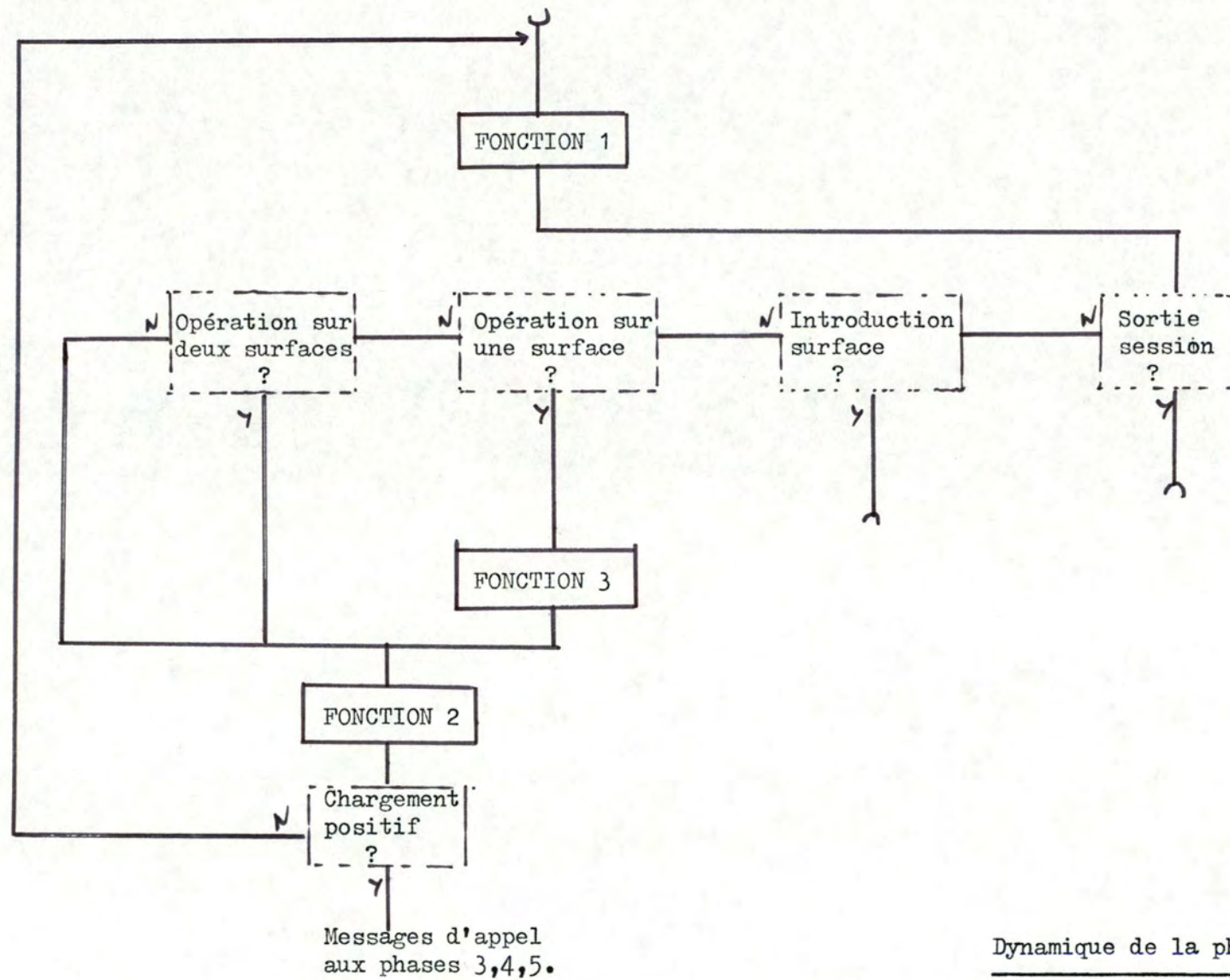
* Fonction 2 : Chargement des données de la surface originale chimique:

Elle est identique à la fonction 2 de la phase 2A.

* Fonction 3 : Initialisations de l'opération sur une surface originale chimique.

Elle est identique à la fonction 3 de la phase 2A, si ce n'est :

- . que les choix de mise à jour de la surface et de sa modification de représentation ne seront pas implémentés. Le choix de suppression d'une surface sera possible malgré tout par le biais d'un appel à la phase 5.
- . que les surfaces originales chimiques seront seulement du type nom.O.



Dynamique de la phase 2C.

A II.5. Phase 3 : Etude d'une surface.

A II.5.1. Sous-schéma des données.

Le sous-schéma de cette phase reprend en fait les sous-schémas des phases 2A, 2B et 2C. On se rapportera donc aux paragraphes A II. 2.1., A II.3.1. et A II.4.1.

A II.5.2. Sous-schéma des traitements.

a. Mise en oeuvre.

Venant du coordinateur où la surface à l'étude fut choisie, on connaît les paramètres essentiels de celle-ci concernant :

* son type de représentation : . EQUATION

ou

. MATRICE D'INTERPOLATION

ou

. MATRICE D'APPROXIMATION

ou

. LISTE de POINTS.

* son identité

* son type : Simple-Valeur ou Multiple-Valeur.

* son origine : . Surface ORIGINALE simple ou géographique ou chimique, associée ou non à une surface approximée ou matricielle.

. MANIPULATION dont la "mère" est une surface originale simple ou géographique ou chimique.

Suivant des différents paramètres, on aura une mise en oeuvre particulière:

a) Si MANIPULATION : On devra reproduire, à partir des données de la surface originale "mère", cette surface manipulée en régénérant les calculs des manipulations subies. En fin de calculs, les coordonnées-écran résultantes de la surface seront disponibles.

b) Si surface ORIGINALE : Un "simple" calcul des coordonnées-écran sera nécessaire.

Ensuite dans les deux cas, on visualisera la surface à l'étude et l'on proposera :

- les différents options d'étude de celle-ci.

* mémorisation de la surface visualisée

* impression de la surface visualisée

* manipulation interactive de la surface visualisée.

* retour à la surface PREMIERE : On entend ici par surface première, la surface non encore modifiée par la phase 3 et issue du premier affichage à l'écran.

- les différentes options de gestion du LOOK de la surface :

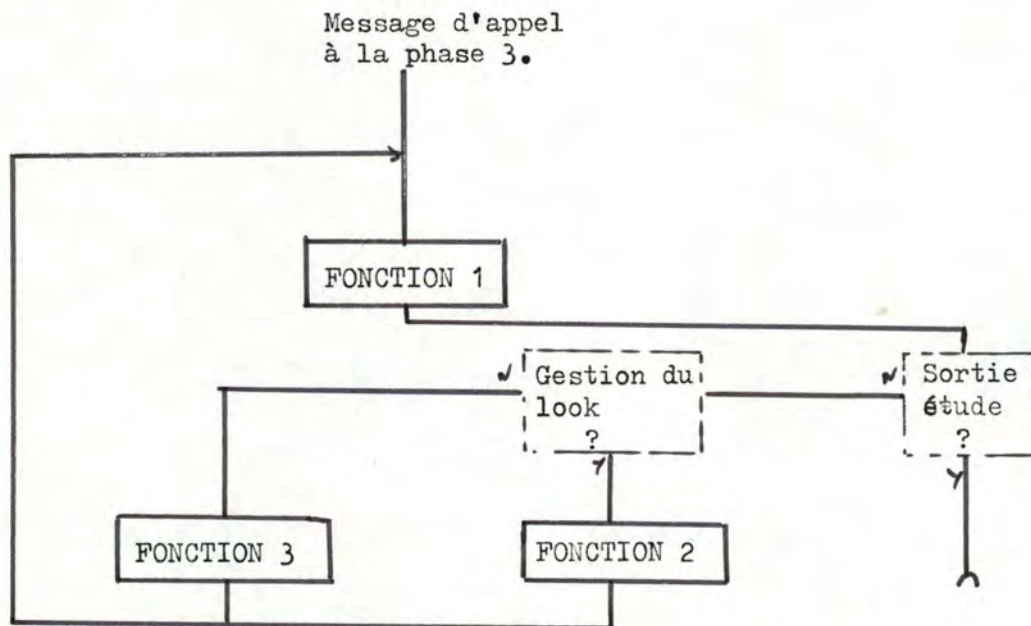
- * visualisation en "vu et caché".
- * affichage permanent de points.
- * affichage permanent des axes X,Y,Z.
- * inversion des coordonnées par rapport au plan (X,Y).
- * rotation continue de la surface.
- * visualisation en bloc-diagramme.
- * modification de l'intensité d'affichage de la surface.
- * visualisation avec isolignes.

b. Découpe en fonction (Voir tableau 2.)

On peut déceler logiquement trois fonctions en rapport avec le type d'action réalisée sur la surface :

- . "gérer l'affichage de la surface à l'étude et coordonner cette étude" (1).
- . "gérer les caractéristiques de visualisation de la surface visualisée".(2).
- . "gérer les manipulations interactives de la surface visualisée"(3)

c. Dynamique des fonctions:



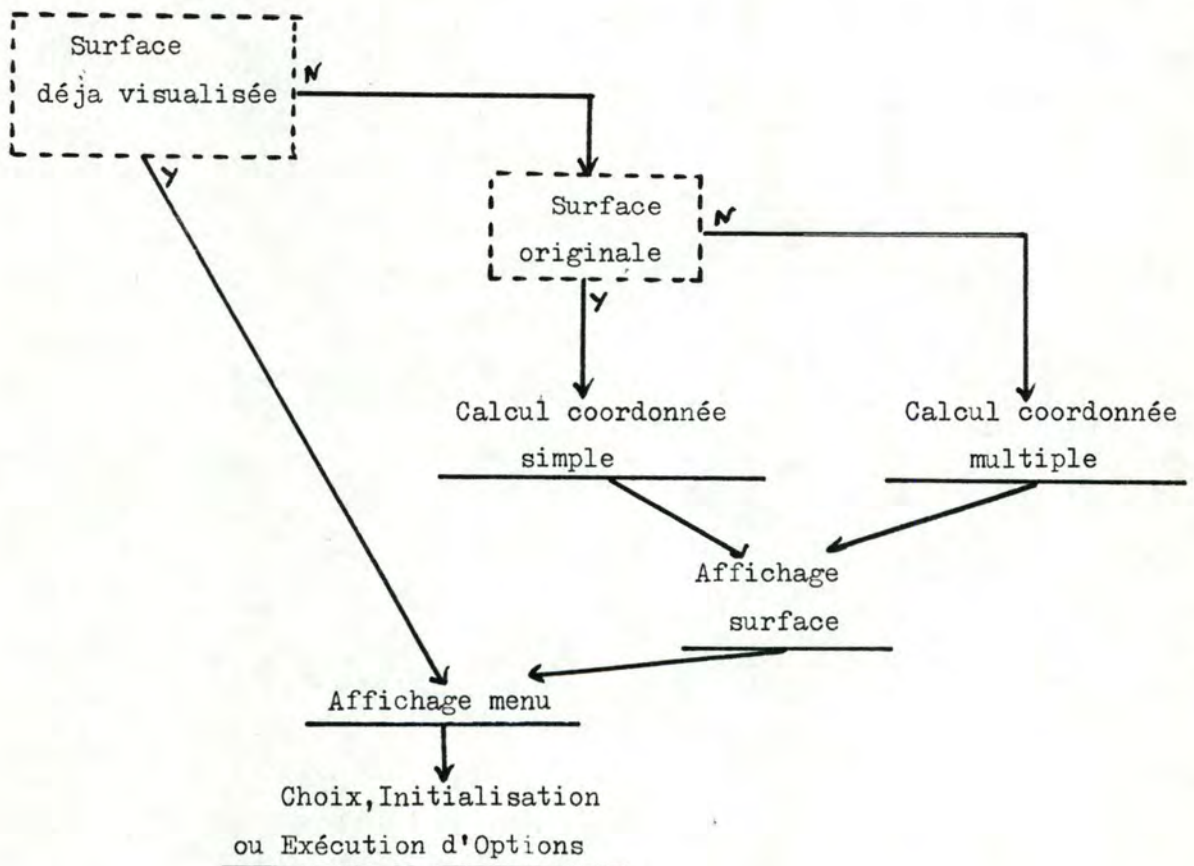
d. Statique des fonctions.

* Fonction 1 : Gestion de l'affichage de la surface à l'étude et Coordination.

- . Condition d'activation : Existence d'un appel valide à la fonction qui correspond à la fin de la fonction 2 ou 3 où un appel venant de la phase 2. Le terme "valide" indique que les messages entrées sont transmis de façon correcte à la fonction 1.
- . Messages entrées :
 - Surface déjà visualisée: - Renseignements généraux lors de la phase 3.
 - Données.
 - Coordonnées-écran de l'état actuel de la surface.

Cela correspondra au message d'appel aux fonctions 1-2" ET, "message" d'appel à la fonction 1.

OU - Surface première : Message d'appel à la phase 3, venant de la phase 2,
- . Traitement : De façon schématique,



. De façon détaillée,

- CALCUL COORDONNEES MULTIPLE : Ceci correspond au calcul non-inter-actif, à partir des données de la surface et des renseignements décrivant les manipulations à reconstituer sur cette surface "mère", des coordonnées-écran de la surface finale manipulée à visualiser.

- CALCUL COORDONNEES SIMPLE : Ceci correspond au simple calcul des coordonnées-écran de la surface originale à visualiser.

- AFFICHAGE SURFACE : Ceci reprend la simple opération de visualisation de la surface à partir des coordonnées-écran disponibles.

- AFFICHAGE MENU : On affichera le choix d'options suivant :

- A MEMORISATION.
- B IMPRESSION.
- C MODIFICATION LOOK.
- D MANIPULATION.
- E RETOUR PREMIERE.
- F SORTIE PHASE.

- CHOIX, INITIALISATION, EXECUTION D'OPTIONS : De façon répétée, l'utilisateur choisira parmi ces options celles qui l'intéresse :

Si A : On créera une nouvelle surface (T.E. MANIPULATION avec NOM. M.(N°+1)), correspondant à la surface actuellement visualisée. Si celle-ci est identique à la surface première, il n'y aura pas de mémorisation possible.

Si B : On réalisera simplement un tirage hard-copy unique de la surface actuellement visualisée. On y inclura le titre de l'écran à ce moment.

Si C : On fera appel à la Fonction 2 en produisant un "message".

Si D : On fera appel à la fonction 3 en produisant un "message".

Si E : On régèrera la surface première si celle-ci n'est pas actuellement à l'écran.

Si F : On fera appel à la Phase 2^{en} produisant un "message"

. Messages Sorties: : Message d'appel aux fonctions 1-2-3 (Renseignements généraux, Données, Coordonnées-écran de la surface)

ET "message" d'appel à la fonction 2 ou 3

OU . "Message" d'appel à la phase 2.

* Fonction 2 : Gestion des caractéristiques de visualisation de la surface visualisée.

- . Objectifs : On offrira la possibilité de modifier(changer,ajouter,supprimer) les caractéristiques de visualisation d'une surface actuellement visualisée.Pour une surface première,ces paramètres de visualisation sont fixés,par défaut :

Type Surface :	Paramètres par défaut :
EQUATION	Wire-Frame,axes lift-honded.
POINTS approximés interpolés	Wire-Frame,axes lift-honded.
POINTS autres	Par points d'intensité variable axes lift-honded.

- . Condition d'activation : Existence d'un appel valide à la fonction 2.
- . Messages entrées : . Message d'appel aux fonctions 1-2-3(Renseignements, généraux,Données,écran).
- . "Message" d'appel à la fonction 2.
- . Traitement : . De façon interactive,l'utilisateur choisira de modifier les caractéristiques de visualisation de la surface.
- Ainsi il pourra :

Type de Modification	Condition d'exécution	Paramètre
A : déclencher la visualisation "Vu et caché"	Simple valeur	Coordonnée
B : déclencher et réaliser l'affichage permanent de points		
C : déclencher et réaliser l'affichage permanent des axes		
D : déclencher et réaliser l'inversion des coordonnées	Simple-valeur	
E : déclencher et réaliser la visualisation en bloc-diagramme.	Simple-valeur	Intervalle
F : déclencher et réaliser la visualisation avec isolignes.	Simple-valeur	
G : déclencher et réaliser la rotation continue de la surface		
H : déclencher et réaliser la modification de l'intensité de la surface.		

Pour chacune de ces options, on devra veiller d'une part à vérifier la condition de déclenchement et d'autre part à introduire les paramètres d'exécution. (Voir tableau ci-dessus).

. L'utilisateur pourra rétablir (ce qui équivaut à une non-validation) le "look" de la surface avant la dernière modification de paramètres de visualisation.

. Messages Sorties : . Message d'appel aux fonctions 1-2-3 (Renseignements généraux, mise à jour, données, coordonnées-écran modifiées).

. "Message d'appel à la fonction 1

. Et éventuellement, "message" d'appel au calcul "VU et CACHE".

* Fonction 3 : Gestion des manipulations de la surface visualisée.

. Objectif : Cette fonction gérera l'ensemble des manipulations interactives d'une surface actuellement visualisée en

- permettant le choix d'une manipulation,
- effectuant celle-ci,
- visualisant le résultat,
- permettant l'éventuel retour à la situation avant la manipulation.

. Condition d'activation : Existence d'un appel valide à la fonction 3.

. Messages entrées : . Message d'appel aux fonctions 1-2-3 (Renseignements généraux, données, coordonnées-écran).

. "Message" d'appel à la fonction 3.

. Traitement : . On affichera tout d'abord les options de manipulation : A - ROTATION X,Y,Z.

B - TRANSLATION X,Y,Z.

C - ACCES A UN POINT.

D - FENETRAGE.

E - COUPE.

F - CHANGEMENT D'ECHELLE.

G - REJET DE LA SURFACE.

H - SORTIE FONCTION.

. On choisira l'option désirée.

. On exécutera l'option choisie :

Si A : rotation de la surface suivant l'axe X,Y ou Z;

Si B : translation de la surface suivant l'axe X,Y ou Z;

Si C : on pointera les points désirés et leurs coordonnées seront affichées. Par un signal d'arrêt, on reviendra au choix d'option.

Si D : On définira le cadre de la fenêtre; possibilité de revenir à la situation avant fenêtrage par l'option G.

Si E : On définira le plan de coupe; la coupe sera alors visualisée en lieu et place de la surface.

On devra alors revenir à la situation avant coupe par l'option G pour réaliser une autre option.

Remarque : Si l'utilisateur veut mémoriser ou imprimer la coupe, il lui sera possible de le faire avant de revenir à la situation avant coupe.

Si F : Scaling de la surface par rapport au centre de l'écran;

Si H : Appel à la FC1 sans affecter la surface visualisée.

Remarque importante :

* Pour les options A,B et F, on pourra directement réaliser ces manipulations par hardware.

* Pour les options C,D et E, on devra auparavant introduire les paramètres de la manipulation.

* L'option G ne sera activable qu'après une manipulation.

- . Messages sorties : . Message d'appel aux fonctions 1-2-3 (Renseignements généraux, données, coordonnées-écran modifiées.
- . "Message" d'appel à la fonction 1.

A.II.6 Phase n°4 : Etude de deux surfaces.

A.II.6.1 Sous-schéma des données.

Le sous-schéma de cette phase reprend en fait les sous-schémas des phases 2A, 2B et 2C.

On se rapportera donc aux paragraphes AII.2.1, AII.3.1 et AII.4.1.

A.II.6.2 Sous-schéma des traitements.

a) Mise en oeuvre.

Elle est proche de celle de la phase 3 (Voir paragraphe AII.4.2.a).

Toutefois, plusieurs différences majeures sont à noter :

- . on travaillera avec deux surfaces,
- . à partir des surfaces originales ou des manipulations reconstituées, on affichera les deux surfaces en des lieux différenciés de l'écran; les surfaces auront la même intensité que leur nom affiché en permanence, et ceci, afin de les identifier.

- . on pourra, pour une surface de travail choisie, réaliser les options d'études identiques à la phase 3, en y ajoutant la possibilité de supprimer l'affichage d'une des deux surfaces.

- . deux options d'étude supplémentaires (comparaison spatiale et statistique) seront proposées.

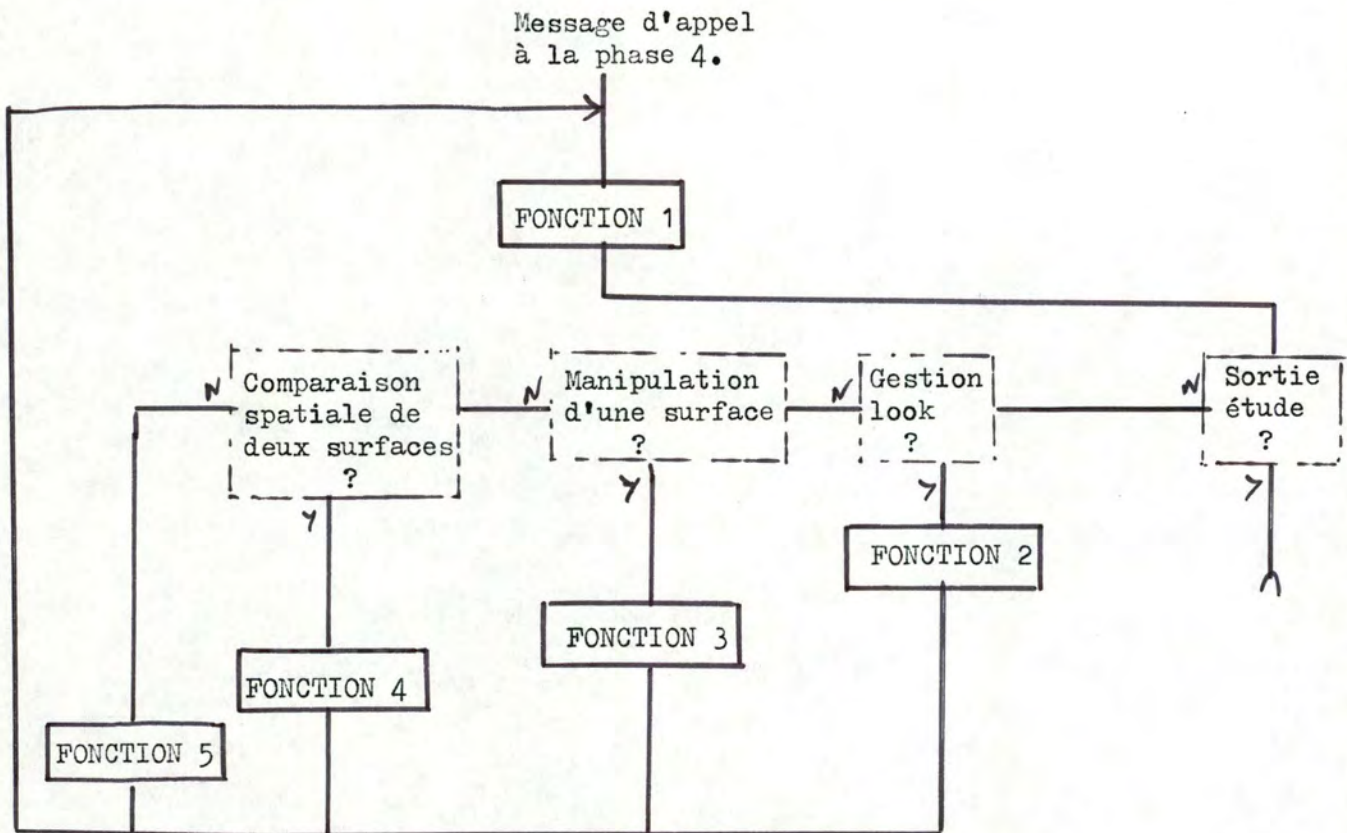
b) Découpe en fonctions (voir tableau 2).

Cette découpe intégrera les trois fonctions de la phase 3 en y ajoutant les actions distinctes de :

- ."gestion d'une comparaison spatiale de deux surfaces" (4)
- ."gestion d'une comparaison statistique de deux surfaces" (5)

c) Dynamique des fonctions.

Voir schéma page suivante.



Dynamique de la phase 4.

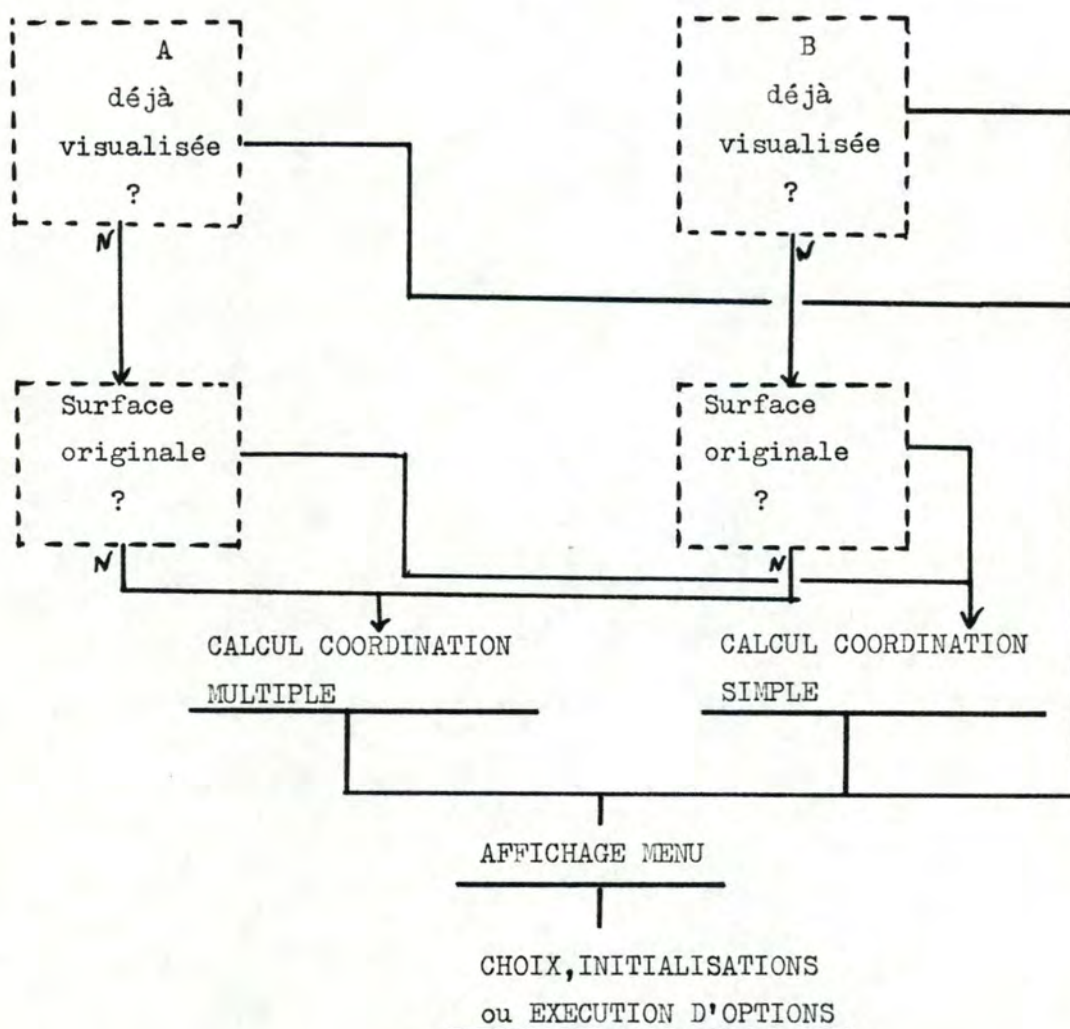
d) Statique des fonctions.

* Fonction 1 : Gestion de l'affichage de deux surfaces (AetB) à l'étude et coordination.

Cette fonction est semblable à la fonction 1 de la phase 3 si ce n'est que :

- . les messages en entrée comprennent soit, deux surfaces déjà visualisées (message d'appel aux fonctions 1,2,3,4,5) et un "message" d'appel à la fonction 1, soit deux surfaces premières (message d'appel à la phase 4).

- . l'algorithme de traitement est le suivant :



. l'affichage du menu comportera les options supplémentaires de suppression d'une surface, de comparaison spatiale (avec production d'un "message" d'appel à la fonction 4) et de comparaison statistique (avec production d'un "message" d'appel à la fonction 5).

* Fonction 2 : Gestion des caractéristiques de visualisation d'une surface visualisée.

Cette fonction est semblable à la fonction 2 de la phase 3 si ce n'est que l'utilisateur devra choisir en plus d'un paramètre de visualisation à modifier, la surface sur laquelle la modification aura lieu.

* Fonction 3 : Gestion des manipulations d'une surface visualisée.

Cette fonction est semblable à la fonction 3 de la phase 3, si ce n'est que l'utilisateur devra choisir en plus d'un paramètre de manipulation, la surface sur laquelle la manipulation portera.

* Fonction 4 : Gestion de la comparaison spatiale de deux surfaces visualisées.

. objectif : A partir de deux surfaces visualisées, on réalisera la soustraction (A-B ou B-A) ou l'addition de celles-ci. La résultante sera visualisée.

De plus, elle sera imprimable ou mémorisable.

. condition d'activation : existence et validité des messages en entrée.

- . messages en entrée :- "message"d'appel à la fonction 4
 - message d'appel aux fonctions 1,2,3,4,5(renseignements,données, coordonnées-écran des 2 surfaces).

. règles de traitement : Tant que l'on reste dans la

fonction 4 : - on choisira tout d'abord la comparaison spatiale désirée (A-B,B-A,A+B)

- on visualisera la résultante de cette comparaison en lieu et place des deux surfaces "mères".

- on pourra alors imprimer ou mémoriser (T.E MANIPULATION tel que le dernier élément de l'élément TYPE-MANIPULATION = comparaison).

- on restituera enfin les deux surfaces "mères" à l'écran.

- . messages en sortie : - message d'appel aux fonctions 1,2,3,4,5.
 - "message"d'appel à la fonction 1.

* Fonction 5 :Gestion de la comparaison statistique de deux surfaces visualisées.

. objectif : Pour les deux surfaces actuellement visualisées, on réalisera une comparaison statistique par la mesure de l'écart moyen entre les deux surfaces suivant une direction. Ce calcul sera fait en batch.

. condition d'activation : existence et validité des messages en entrée.

- . messages en entrée : -"message" d'appel à la fonction 5
 - message d'appel aux fonctions 1,2,3,4,5.

. règles de traitement : Tout en conservant les deux surfaces visualisées, on introduira la direction de mesure de l'écart moyen par le biais de deux coordonnées de points dans l'espace réel. On engendrera ensuite un message d'appel au calcul "comparaison-statistique" reprenant les caractéristiques des surfaces et la direction de mesure d'écart.

- . messages en sortie : - message d'appel aux fonctions
1,2,3,4,5.
- "message" d'appel à la fonction 1.
- "message" d'appel au calcul
"comparaison-statistique".

A.II.7 Phase n°5 : Introduction, consultation et mise à jour d'une surface.

A.II.7.1 Sous-schéma des données.

le sous-schéma de cette phase reprend en fait les sous-schémas des phases 2A, 2B et 2C.

On se rapportera donc aux paragraphes AII.2.1; AII.3.1 et AII.4.1.

A.II.7.2 Sous-schéma des traitements.

a) Mise en oeuvre.

Au coordinateur (phase n°2), on a déjà pris l'option d'une création de surface (1) ou d'une consultation et mise à jour (2).

- * Si (1) : on traitera en premier lieu les renseignements et puis les données de la surface. Dans les deux cas, après introduction d'un ENSEMBLE COMPLET de VALEURS, c'est-à-dire une liste de points

ou

une ou plusieurs équations

ou

un ensemble de renseignements,

on effectuera les actions suivantes :

- . IMPRESSION : les valeurs validées sont imprimées,
- . VALIDATION : les valeurs sont mémorisées,
- . FIN D'INTRODUCTION : on quitte la session d'introduction
avec ou sans validation,
- . RECTIFICATION : les valeurs ne sont pas mémorisées et on recommence une session d'introduction.

Si la surface est de type P : on introduira les données au clavier
 ou
 à la table de digitalisation
 ou
 par un fichier extérieur.

Si la surface est de type E : on introduira au clavier.

- * Si (2) : on traitera également les renseignements puis les données. Au départ, il sera également possible de supprimer toute une surface (données et renseignements)
 Pour chaque ensemble complet de valeurs choisi, on affichera les valeurs existantes et par curseur, on opérera (ajouter, lire, modifier, supprimer) sur les items de cet ensemble.

Ensuite, on effectuera les actions suivantes :

- . VALIDATION : les mises à jour sont mémorisées,
- . IMPRESSION : les valeurs mises à jour sont imprimées,
- . FIN de MISE A JOUR : on quitte la session de mise à jour
 avec ou sans validation,
- . RECTIFICATION : on réitère une session de mise à jour sans avoir validé la précédente.

b) Découpe en fonctions (tableau 2).

Cette phase s'articule autour de deux axes.

En effet, selon la mise en oeuvre, on distingue l'introduction d'une surface et la mise à jour ou consultation des données et renseignements de la surface.

Logiquement, au sein de ces deux axes, on devra gérer l'introduction ou la consultation - mise à jour d'une part des renseignements, d'autre part, des données originales de la surface.

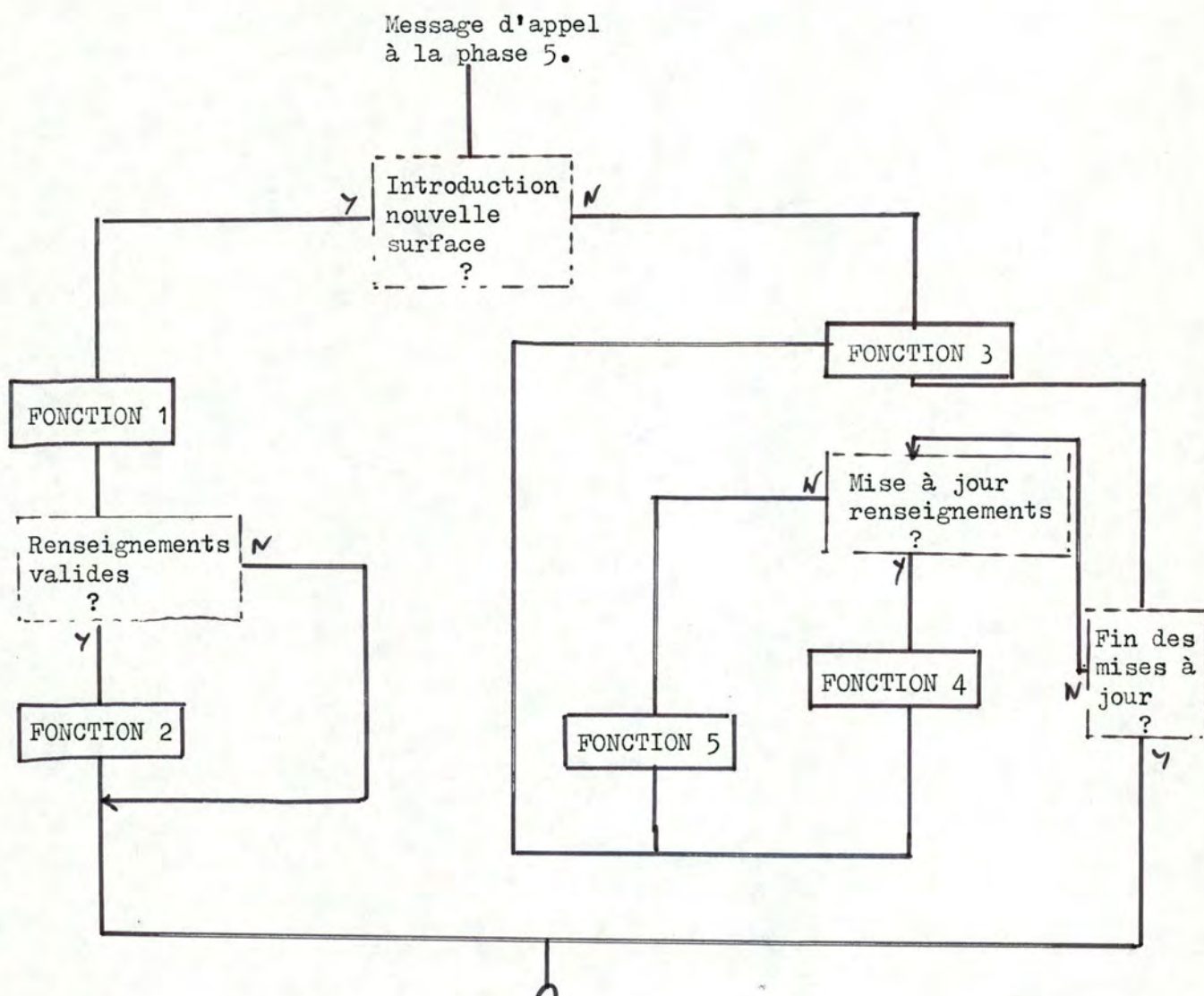
L'introduction des données et des renseignements concernant une surface était liée par nature (ces deux introductions se rapportent au même phénomène), ces deux étapes se devaient d'être successives et interdépendantes; d'où une phase de coordination entre les deux étapes s'avérait inutile.

La consultation - mise à jour se doit, elle, d'être plus souple et l'on a donc placé un mini coordinateur gérant les étapes concernant d'une part, les renseignements, d'autre part, les données.

En conclusion, on a les actions élémentaires suivantes :

- "gérer l'introduction des renseignements généraux d'une surface" (1).
- "gérer l'introduction des données d'une surface" (2).
- "coordonner la consultation - mise à jour d'une surface" (3).
- "gérer les consultations - mises à jour des renseignements généraux d'une surface" (4).
- "gérer les consultations - mises à jour des données d'une surface" (5).

c) Dynamique des fonctions.



d) Statique des fonctions.

* Fonction 1 : Introduction des renseignements généraux d'une surface.

. condition d'activation : existence d'un "message" d'appel à la phase 5.

. message en entrée : "message" d'appel à la phase 5 (pour introduction d'une surface).

. règles de traitement : - on affichera à l'écran les trames des renseignements généraux à introduire : voir tableau page suivante.

- le curseur du clavier se déplacera de ligne en ligne.

Pour chaque ligne, on pourra déplacer le curseur afin de gérer son introduction.

On procédera enfin à la validation de la ligne.

Les renseignements soulignés devront être obligatoirement introduits.

Les renseignements TYPE-CENTRE et COORDONNEES-CENTRE seront obligatoires si le TYPE-DONNEES = 'P' ou 'P.D'.

- quand un ensemble complet de données est introduit, on pourra choisir de :

. le valider et donc de transmettre l'ensemble à la fonction 2.

On engendrera le "message" d'appel à la fonction 2.

. le rectifier et donc de réinitialiser l'introduction au début de la fonction 1.

. ne pas continuer la phase d'introduction.

On engendrera un "message" d'appel à la phase 2.

. l'imprimer et de le valider.

On engendrera le "message" d'appel à la fonction 2.

. message en sortie : "message" d'appel à la fonction 2 et un message d'appel à la fonction 2 (contenant les renseignements validés) ou "message" d'appel à la phase 2.

* Fonction 2 : Introduction des données d'une surface.

. condition d'activation : existence d'un "message" d'appel à la fonction 2 et d'un message d'appel à la fonction 2.

Version simple	Version géographique	Version chimique
<u>NOM</u> : - - - - - . 0 <u>TYPE-DEFINITION</u> (P,E) : - . <u>SENS-AXES</u> (LH,RH) : - - . <u>TYPE-SURFACE</u> (SV,MV) : - - . <u>CREATEUR</u> : - - - - -	<u>NOM</u> : - - - - - . 0 <u>TYPE-DEFINITION</u> (P) : P. <u>SENS-AXES</u> (LH,RH) : - - . <u>TYPE-SURFACE</u> (SV) : SV. <u>CREATEUR</u> : - - - - - <u>LIEU</u> : - - - - - <u>TYPE-CENTRE</u> (I,C) : - <u>COORDONNEES-CENTRE</u> : (- - - - -); (- - - - -);(- - - - -). <u>TYPE-DONNEES</u> (P,D,PD) : - - .	<u>NOM</u> : - - - - - . 0 <u>TYPE-DEFINITION</u> : - . <u>SENS-AXES</u> (LH,RH) : - - . <u>TYPE-SURFACE</u> (MV) : MV. <u>CREATEUR</u> : - - - - - <u>NOMBRE-ATOMES</u> : - - - - - <u>DENSITE-REQUIRE</u> : - - - - - <u>NOMBRE-TYPE-ATOMES</u> : - - - - - <u>RAYON-SONDAGE</u> : - - - - -

. message en entrée : "message"d'appel à la fonction 2 et message d'appel à la fonction 2.

. règles de traitement :- suivant le type de définition ('P' ou 'E'), on pourra choisir entre :

* si 'P' : *introduction par un fichier (a)
 *introduction au clavier (b)
 *introduction à la table de digitalisation (si type surface = 'SV') (c)

* si 'E' : *introduction par un fichier (a)
 *introduction au clavier (d)

_ Si (a) : on introduira simplement le nom du fichier.

Après réalisation de cette introduction, on pourra imprimer le fichier et sortir de la phase ("message"d'appel à la phase 2).

_ Si (b) : on affichera la trame d'introduction de points :

$\begin{cases} X = - - - - - \\ Y = - - - - - \\ Z = - - - - - \end{cases}$	et si dans la version géogra- phique, le type- données = P.D	$\begin{cases} X = - - - - - \\ Y = - - - - - \\ Z = - - - - - \end{cases}$
---	---	---

on introduira alors les points; chaque point fera l'objet d'une validation.

Une fois un ensemble de points introduits, on pourra alors choisir entre les mêmes options que celles présentées à la fonction 1 (valider, rectifier, valider et imprimer, stopper).

_ Si (c) : on introduira tout d'abord les spécifications du plan qui correspondra à la table de digitalisation : X = ou Y = ou Z =

on introduira ensuite un point en déplaçant la souris sur la table et en spécifiant la troisième coordonnée par le clavier.

Chaque point fera l'objet d'une validation de ses coordonnées. L'ensemble des points introduits, on pourra alors choisir les options offertes par la fonction 1.

_ Si (d) : on introduira le nombre et le type d'équations.

Dans un premier temps, nous permettrons l'introduction d'équation cartésienne de surfaces du premier degré ($AX + BY + CZ + D = 0$) et du second degré ($AX^2 + BY^2 + CZ^2 + 2DXY + 2EXZ + A'X + B'Y + 2F'YZ + C'Z + G = 0$).

Suivant ces paramètres, on affichera la trame d'introduction

d'équations : . $A(X) =$ $B(Y) =$ $C(Z) =$ $D =$
 . $A(X^2) =$ $B(Y^2) =$ $C(Z^2) =$ $D(XY) =$ $E(XZ) =$
 $F(YZ) =$ $A'(X) =$ $B'(Y) =$ $C'(Z) =$

On introduira alors les coefficients des équations.

Chaque équation fera l'objet d'une validation.

Une fois un ensemble complet d'équations introduites, on proposera les options de la fonction 1.

. messages en sortie : "message" d'appel à la phase 2.

* Fonction 3 : Coordination des mises à jour d'une surface.

. condition d'activation : Existence d'un message d'appel à la phase 5; ou d'un message d'appel aux fonctions 3,4,5 et un "message" d'appel à la fonction 3.

. message en entrée : message d'appel à la phase 5; ou message d'appel aux fonctions 3,4,5 (contenant les données et renseignements modifiés de la surface) et "message" d'appel à la fonction 3.

. règles de traitement :-on choisira entre les options suivantes : .fin des consultations-mises à jour (1)

.suppression d'une surface (2)

.consultation-mise à jour des renseignements (3)

.consultation-mise à jour des données (4)

.impression des renseignements et/ou données.

avec les conditions d'application suivantes :

. la suppression d'une surface devra subir une double validation.

. la consultation-mise à jour des renseignements est seulement permise aux surfaces originales (nom.0) et aux surfaces modélisées géographiques(nom.M.GEO).

Seule une modification des renseignements n'entraînant pas une modification de la surface sera possible : NOM, SENS-AXES, CREATEUR TYPE-CENTRE, COORDONNEES-CENTRE, si le TYPE-DONNEES = 'D'.

. la consultation-mise à jour des données est interdite aux surfaces chimiques et seulement permises aux surfaces originales (nom.0).

. la modification de points ou d'équations entraînera la création d'une autre surface (identique à celle de base sauf au niveau des modifications) ou, selon le choix de l'utilisateur, le remplacement de la surface de base par la surface modifiée. Le second principe entraînera la possibilité d'invalidation des renseignements généraux (la surface sera-t-elle toujours simple valeur? par exemple) et aussi des manipulations qui étaient issues de la surface de base.

L'utilisateur devra donc être prudent.

. l'impression pourra être accessible à tous les types de surfaces.

: - après validation du choix, on engendrera les messages d'appel.

. message en sortie : Si (1) : "message"d'appel à la phase 2.

Si (2) : "message"d'appel à la phase 2.

Si (3) : "message"d'appel à la fonction 4 et message d'appel aux fonctions 3,4,5.

Si (4) : "message"d'appel à la fonction 5 et message d'appel aux fonctions 3,4,5.

* Fonction 4 : Mise à jour des renseignements généraux d'une surface.

. condition d'activation : existence d'un "message" d'appel à la fonction 4 et d'un message d'appel aux fonctions 3,4,5.

. messages en entrée : "message"d'appel à la fonction 4 et message d'appel aux fonctions 3,4,5.

. règles de traitement : - on affichera à l'écran les trames garnies des renseignements généraux à modifier (voir fonction 1).

: - le curseur du clavier se déplacera de ligne en ligne afin de modifier les items modifiables (voir fonction 3).

: - en fin de mise à jour, on choisira entre les options suivantes :

- . valider les modifications et sortir de la fonction 4 (1)
- . réinitialiser les modifications sans validation.
- . sortir de la fonction 4 sans validation (2).
- . messages en sortie : Si (1) : "message"d'appel à la fonction 3 et message d'appel aux fonctions 3,4,5 (modifié).
- Si (2) : "message"d'appel à la fonction 3 et message d'appel aux fonctions 3,4,5.

* Fonction 5 : Mise à jour des données d'une surface.

- . condition d'activation : existence d'un "message" d'appel à la fonction 5 et d'un message d'appel aux fonctions 3,4,5.
- . messages en entrée : "message"d'appel à la fonction 5 et message d'appel aux fonctions 3,4,5.
- . règles de traitement : - on affichera la liste des points ou équations de la surface.

: - le curseur du clavier se déplacera dans cette liste afin de modifier les éléments de la liste.

: - en fin de mise à jour, on choisira entre les options mentionnées à la fonction 4.

- . messages en sortie : Si (1) : "message"d'appel à la fonction 3 et message d'appel aux fonctions 3,4,5 (modifié).
- Si (2) : "message"d'appel à la fonction 3 et message d'appel aux fonctions 3,4,5.

A.II.8 Phase n°6 : Modification de Représentation.

A.II.8.1 Sous-schéma des données.

Le sous-schéma de cette phase reprend en fait les sous-schémas des phases 2A,2B et 2C.

On se rapportera donc au paragraphe AII.2.1, AII.3.1 et AII.4.1.

A.II.8.2 Sous-schéma des traitements.

a) Mise en oeuvre.

A partir des données et des renseignements généraux de la surface (type de définition = 'P', type de surface = 'SV'), on choisira la méthode et ses paramètres :

Méthode	Paramètres
Beguin	taille de la maille rayon d'interpolation
Jouret	nombre de voisins intervenant dans le calcul d'un coin de maille, nombre de voisins intervenant dans le calcul du plan tangent à un point (X_i, Y_i) taille de la maille en X et en Y échelle
Brunet	La méthode semble-t-il, peut calculer ses paramètres à partir des points de la surface à approximer.

Après validation des choix, les calculs seront effectués.

Les résultats seront produits sur l'écran.

Au vu des résultats, l'utilisateur choisira entre les options suivantes :

- . validation de la modification de représentation.

Ceci entraînera la mémorisation des données et renseignements de la surface (nom.O.A.NO ou nom.O.M.NO).

L'utilisateur devra choisir s'il remplace la surface originale par la surface originale approximée ou matricielle. Si oui, les occurrences de l'association DEFINIE-PAR-2 et les occurrences de l'entité POINT liées à la surface originale seront supprimées.

On pourra aussi imprimer les résultats.

- . non-validation et réinitialisation de l'introduction des choix de méthodes et paramètres.

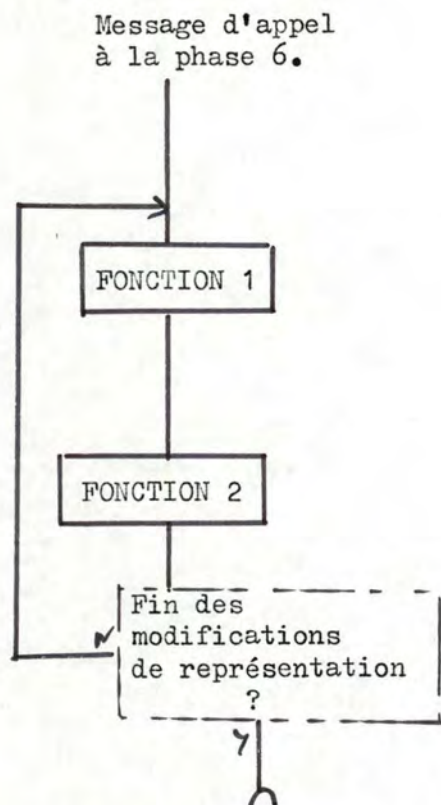
- . fin de la phase de modification.

- b) Découpe en fonctions (voir tableau 2).

La découpe fonctionnelle de cette phase peut suivre le critère d'interactivité. On identifiera alors deux fonctions élémentaires :

- . "gérer la mise en oeuvre de la modification de représentation"(1)
- . "calculer la modification de représentation"(2)

c) Dynamique des fonctions.



d) Statique des fonctions.

* Fonction 1 : Gestion de la mise en oeuvre de la modification de représentation.

. condition d'activation : existence d'un message d'appel à la phase 6 ou d'un message d'appel à la fonction 1.

. messages en entrée : message d'appel à la phase 6 ou message d'appel à la fonction 1.

. règles de traitement :- on présentera les choix des méthodes (Beguin, Jouret, Brunet).

Après validation, on présentera les trames vides de paramètres (pour la méthode choisie) à remplir (voir mise en oeuvre).

:- on fera, après validation des paramètres, appel à la fonction 2 par la production d'un message

d'appel à la fonction 2.

Ce message contiendra les données de la surface et les paramètres de la modification de représentation à apporter.

- . messages en sortie : message d'appel à la fonction 2.

* Fonction 2 : Calcul des traitements d'approximation et d'interpolation.

- . condition d'activation : existence message d'appel à la fonction 2.

- . message en entrée : message d'appel à la fonction 2.

- . règles de traitement :- A partir des données du message d'appel, on calculera les matrices d'interpolation ou d'approximation selon la méthode et ses paramètres choisis.

- A la fin des calculs,

l'utilisateur pourra choisir entre :

- .valider, mémoriser et remplacer l'ancienne représentation de la surface par la nouvelle,

- .valider, mémoriser la nouvelle représentation,

- .invalider le calcul.

- Après une validation,

l'utilisateur pourra imprimer les données et renseignements de la nouvelle représentation.

- Enfin, l'utilisateur pourra sortir de la phase (1), ou réinitialiser la phase (2) si l'ancienne représentation n'a pas été supprimée

- . messages en sortie : Si (1) : "message" d'appel à la phase 2.

- Si (2) : message d'appel à la fonction 1 contenant la représentation de base de la surface (non modifiée).

A.II.9 Phase n°7 : Aide à l'utilisateur.

A.II.9.1 Sous-schéma des données.

Cette phase particulière dans ses objectifs (non directement liés à l'étude de surfaces 3-D), dans sa dynamique (voir paragraphe III.2.2), utilise des données permanentes également particulières.

* Entité INTERACTION : Une interaction correspond à un état du logiciel où l'utilisateur pourra faire appel à la fonction d'aide afin d'obtenir des précisions sur cet état, son passé et son futur possible.

* Elément NO-INTERACTION : Cet élément identifie chaque interaction au sein d'une fonction d'une phase du projet.

Format : 3 numériques . 3 numériques . 3 numériques .
Les trois parties de cet élément correspondent respectivement au numéro de phase (voir tableau 2), au numéro de fonction dans la phase (voir tableau 2) et au numéro d'interaction.

* Entité PAGE-DOCUMENTATION : Associée aux interactions, cette page de documentation décrira les renseignements nécessaires à l'utilisateur.

* Elément NO-PAGE : Cet élément correspond simplement à l'index de référence de chaque page. Format : 4 numériques.

* Elément TEXTE-DOCUMENTATION : Cet élément contiendra les renseignements documentaires. Format : 128 alphanumériques.

* Elément TEXTE-MENU : Cet élément listera les options disponibles au cours d'une interaction. Format : 128 alphanumériques.

A.II.9.2 Sous-schéma des traitements.

a) Mise en oeuvre.

Dans les situations d'attente de dialogue (l'utilisateur doit interagir avec le programme qui attend son action), on pourra réaliser l'appel à la phase d'aide par la commande HELP.

Celle-ci fournira alors les renseignements de base suivants :

- . niveau de phase atteint,
- . niveau de fonction atteint,
- . dernière commande effectuée,
- . commandes futures possibles avec explication de la sémantique et de la syntaxe de ces commandes.
- . données chargées dans le système et à l'étude (nom des surfaces).

Après appel, on retournera dans l'état de l'interaction où a eu lieu l'appel.

Un utilisateur privilégié pourra gérer la documentation (ajout, suppression, modification de pages) lors d'un appel à la phase.

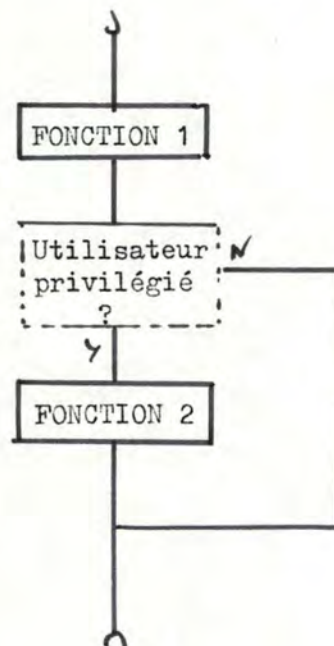
Remarquons que nous n'avons qu'ébauché le problème de la gestion d'un documentaire.

Cette problématique pourrait faire l'objet d'un travail complet concernant la création, la tenue, l'appel d'une B.D documentaire. Ce n'est pas notre but ici.

b) Découpe en fonctions (voir tableau 2).

Deux fonctions nous semblent nécessaires à la mise en oeuvre de la phase afin de :

- . "gérer l'appel HELP"
 - . "gérer la documentation".
- c) Dynamique des fonctions.



d) Statique des fonctions.

* Fonction 1 : Gestion de l'appel HELP.

. condition d'activation : existence d'un "message" d'appel à la phase 7 contenant le numéro de l'interaction où l'appel s'est produit, la dernière commande réalisée et le nom de la surface à l'étude.

. message en entrée : "message"d'appel à la phase 7.

. règles de traitement :- on affichera les pages de documentation liées aux informations du "message" d'appel.

- on reviendra, à la demande, à l'interaction appelante ou, si l'utilisateur est privilégié, on gèrera à la demande la documentation en activant la fonction 2.

. message en sortie : "message"d'appel à l'interaction appelante ou "message"d'appel à la fonction 2.

* Fonction 2 : Gestion de la documentation.

. condition d'activation : existence d'un "message" d'appel à la fonction 2.

. message en entrée : "message"d'appel à la fonction 2.

. règles de traitement :- on choisira tout d'abord si l'on veut, ajouter des pages concernant l'interaction

.supprimer des pages concernant l'interaction

.modifier des pages concernant l'interaction

après avoir affiché la liste des pages liées à l'interaction.

:- on choisira alors le numéro de la page que l'on veut ajouter, supprimer ou modifier.

:- après validation de l'opération, on aura la possibilité de :

. sortir de la phase HELP et de revenir à l'interaction appelante

. choisir à nouveau un type d'opération.

. message en sortie : "message"d'appel à l'interaction appelante.

A.II.10 Phase n°8 : Terminaison d'une session.

A.II.10.1 Sous-schéma des données.

Le sous-schéma de cette phase ne diffère pas des sous-schémas exposés lors des phases 2^A, 2^B et 2^C.

On se rapportera donc aux paragraphes AII.2.1, AII.3.1 et AII.4.1.

A.II.10.2 Sous-schéma des traitements.

a) Mise en oeuvre.

. Dans le cas d'une entrée positive dans une session, on produira les résultats (surfaces mémorisées, surfaces imprimées) de la session et on choisira de sortir ou non du logiciel :

Si oui, on lancera éventuellement les travaux batch de calcul d'une visualisation en "vu et caché" ou de calcul d'une comparaison statistique.

Enfin, on clôturera la connexion au système.

Si non, on choisira la version future de travail et initialisera celle-ci.

. Dans le cas d'une entrée négative, on clôturera la connexion au système.

b) Découpe en fonctions (Voir tableau 2).

Selon les notions de session (ensemble d'interactions réalisées dans le cadre d'une version) et de connexion (ensemble d'interactions réalisées dans le cadre du LGI), nous avons distingué deux actions de base :

- . "Terminer une session" (1)
- . "Terminer une connexion" (2)

c) Dynamique des fonctions.

Voir schéma de la dynamique page suivante.

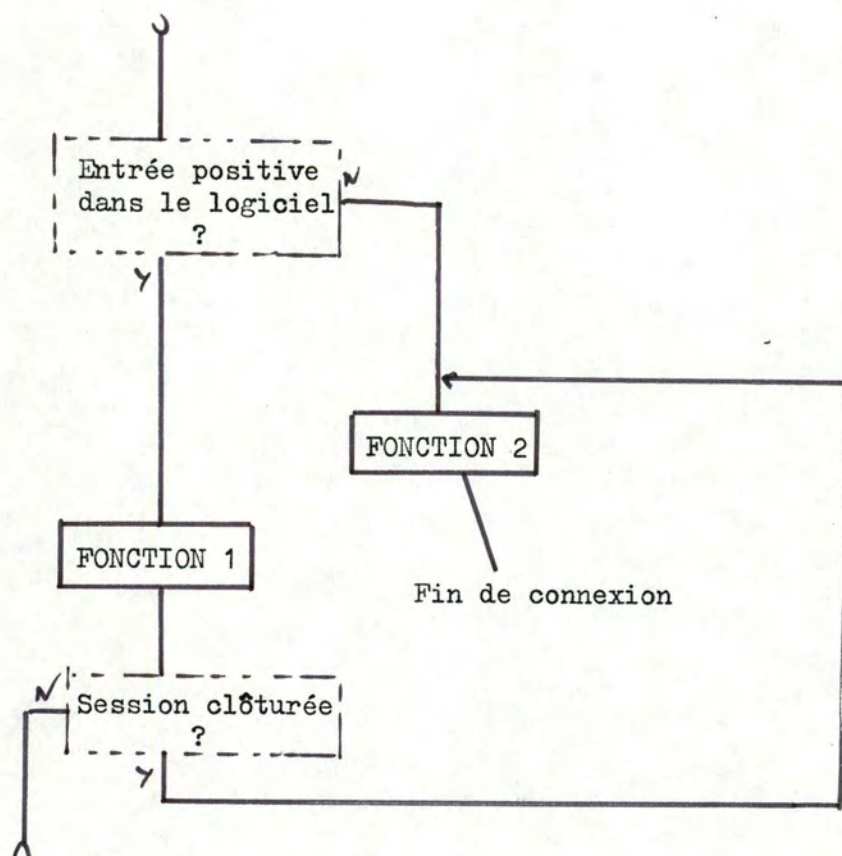
d) Statique des fonctions.

* FONCTION 1 : Terminaison d'une session.

. condition d'activation : existence d'un "message" d'appel à la phase 8.

. message en entrée : "message" d'appel à la phase 8.

. règles de traitement : . on produira à l'écran les résultats en termes de surfaces mémorisées et/ou imprimées.



Dynamique de la phase 8.

. on choisira ensuite de poursuivre éventuellement la connexion :

Si oui, on choisira la version et on initialisera une nouvelle session en produisant un message d'appel à la phase 1 (message d'identification).

Si non, on fera appel à la fonction 2.

. messages en sortie : . message d'identification
ou . "message" d'appel à la fonction 2.

* FONCTION 2 : Terminaison d'une connexion au logiciel

. condition d'activation : existence d'un "message" d'appel à la fonction 2 ET éventuellement, d'un "message" d'appel au calcul "vu et caché" OU "comparaison statistique",

OU existence d'un "message" négatif d'entrée.

. messages en entrée : . message d'appel à la fonction 2 ET, éventuellement, d'un "message" d'appel au calcul "vu et caché" ou "comparaison statistique",

OU . "message" négatif d'entrée.

. règles de traitement :

. Si "message" négatif d'entrée alors, on clôturera l'exécution du logiciel, sinon, si "message" d'appel au calcul "vu et caché" OU "comparaison statistique", on réalisera ces traitements à partir des arguments affectés respectivement lors des phases 3 et 4 (fonction 2) et phase 4 (fonction 5). En fin de traitement, on effectuera les terminaisons non-interactives (fermeture de fichiers,...).

. message en sortie : fin d'exécution du logiciel.

A.II.11 Phase n°9B : Traitements Complémentaires-Version géographique.

AII.11.1 Sous-schéma des données.

Le sous-schéma de cette phase ne diffère pas des sous-schémas exposés lors des phases 2A, 2B et 2C.

On se rapportera donc aux paragraphes AII.2.1, AII.3.1 et AII.4.1.

AII.11.2 Sous-schéma des traitements.

a) Mise en oeuvre.

Cette phase regroupe deux types de traitements distincts et activables de façon indépendante.

Rappelons que ces traitements portent sur le même type de données, ce qui a justifié leur regroupement en une phase.

Ces traitements sont : . Une modélisation interactive d'une variation réelle des densités mesurées par les populations (type-données = 'P') ou par les densités (type-données = 'D').

A partir des données de la surface originale géographique et des paramètres de modélisation (modèle quadratique, newtonien, directionnel), coordonnée du centre, méthode de modélisation (à partir des $D(r)$, à partir des $P(r)$), on créera une surface originale géographique modélisée.

: . Un calcul des résidus d'un modèle par rapport à sa variation réelle.

Ici, les données originales du modèle (équation) et de la variation réelle seront comparées afin de produire la "surface" des résidus (points).

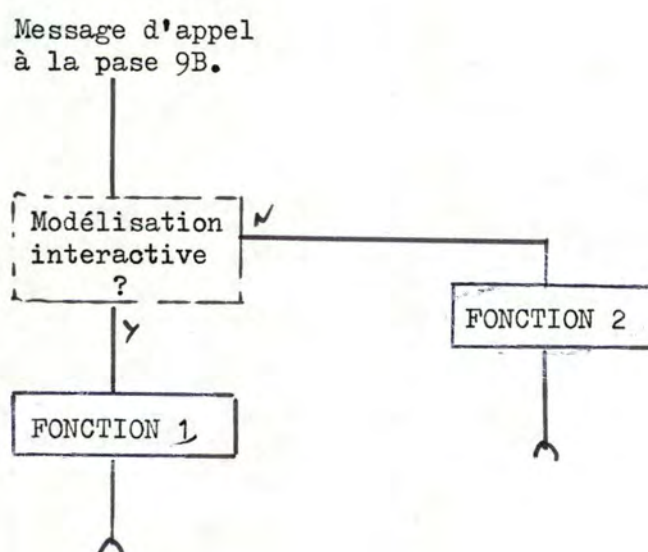
Afin de conserver notre principe d'enregistrement permanent des seules données originales, cette "surface" des résidus ne sera pas permanente et sera donc perdue en fin de connexion.

b) Découpe en fonctions (voir tableau 2).

De façon logique, nous avons identifié deux fonctions au sein de cette phase :

- . "Modéliser interactivement une variation réelle de densités"(1)
- . "Calculer les résidus entre un modèle et sa variation de base (2)

c) Dynamique des fonctions.



d) Statique des fonctions.

* Fonction 1 : Traitements complémentaires à une modélisation

- . condition d'activation : existence d'un message d'appel à la phase 9B et d'un "message" d'appel à une modélisation interactive.
- . message en entrée : message d'appel à la phase 9B.
- . règles de traitement : . on introduira au choix
 - . le modèle à ajuster à la variation réelle,
 - . la méthode de modélisation,
 - . les coordonnées du centre à partir duquel la modélisation sera réalisée.

: . Après validation, par la méthode des moindres carrés, on calculera les coefficients de l'équation du modèle choisi.

: . Les résultats de la modélisation seront produits : - coefficient de détermination
- coefficients de l'équation

: . On choisira alors entre :

- mémoriser la surface modélisée
- invalider les calculs

: . Puis, on décidera de continuer ou d'arrêter la phase de modélisation.

- . messages en sortie : . "message" d'appel à la phase 2B
ET "message" d'appel à la fonction 3 de la phase 2B.

* Fonction 2 : Traitements complémentaires à un calcul de résidus

- . condition d'activation : existence d'un message d'appel à la phase 9B et d'un "message" d'appel à un calcul de résidus.
- . message en entrée : message d'appel à la phase 9B.
- . règles de traitement : . A partir des données originales de base de la surface et de l'équation du modèle, on calculera les écarts au niveau du maillage de l'approximation ou d'interpolation.

. On produira ensuite les résultats ainsi que la possibilité d'imprimer les résidus.

Cette "surface" des résidus aura pour identifiant le NOM.M.GEO.NO.R.
Elle ne fait pas partie des données permanentes de la phase, mais
pourra être utilisée dans la phase 4 aux fins de comparaison avec
la variation réelle.

- messages en sortie : • "message" d'appel à la phase 2B
ET • "message" d'appel à la
fonction 4 de la phase 2B.
-

ANNEXE III

Spécifications abstraites des modules de l'architecture physique.

A III.0 Remarques préliminaires.

Dans le cadre des restrictions présentées au chapitre IV (§ IV.6), nous allons au préalable décrire les modules nécessaires pour atteindre les objectifs de base cités.

Ces restrictions, en vertu des propriétés d'une architecture logicielle basée sur la relation UTILISE, devrait être possible et nous mener malgré tout à un ensemble cohérent. Après restrictions trois actions devraient être encore possibles:

- "Introduire une surface définie par des points ou une équation, en simple-valeur". Les modules n° 615, 616 et 617 seront alors utiles ainsi que le module de données n° 41 et les modules de gestion des devices (n° 44 pour la saisie au clavier, n° 416 pour l'affichage des données au terminal, n° 415 pour la saisie des données à la table).

- "Visualiser la surface". Cette opération nécessitera les modules n° 53 (pour le calcul des coordonnées-écran de la surface) et n° 413 (pour l'affichage).

- "Manipuler interactivement la surface". Ceci entraînera la mise en oeuvre des modules n° 69 et 611 (pour la gestion des manipulations), n° 410 (pour l'affichage des commandes disponibles), n° 412 (pour l'introduction des commandes), n° 413 (pour l'affichage de la surface), n° 414 (pour la gestion de certaines commandes), et n° 53 (pour le calcul de la surface manipulée).

En définitive, les modules n° 69, 611, 615, 616, 617, 53, 41, 44, 410, 412, 413, 414, 415 et 416 seront spécifiés abstraitement dans cette annexe III.

A III.1 Module n° 69 : Initialisations étude une surface.

Remarque importante : Dans notre texte, nous avons placé en majuscules l'expression des TYPES de données permanentes reprises à la figure 26. Les occurrences elles seront écrites en minuscules.

• Arguments : .Nom de surface = (NOM-SURFACE.O) ou (NOM-SURFACE.O.A.NO) ou (NOM-SURFACE.O.M.NO) ou (NOM-SURFACE.M.NO) ou (NOM-SURFACE.M.GEO.NO).

.Type d'opération sur la surface = (MAJ) ou (MODIF) ou (ETUDE 1) ou (ETUDE 2) ou (NEW) ou (OUT).

.Renseignements généraux de la surface =

+ renseignements liés à une version.

-Si NOM-SURFACE.O alors (nom-surface.o,type-définition,sens-axes,type-surface,nom-créateur) (1)
OU éventuellement, si } NOM-SURFACE.O.A.NO alors ((1),méthode d'approximation).
} NOM-SURFACE.O.M.NO alors ((1),méthode de matricialisation,taille-maille,taille-cercle).
-Si NOM-SURFACE.M.NO et nom-surface-origine est unique alors ((1),nom-surface-origine,(2) si } ,
type-manipulation).
sinon (((1),nom-surface-origine 1 ,(2) si })et
((1),nom-surface-origine 2 ,(2) si })et
(type-manipulation)).
-Si NOM-SURFACE.M.GEO alors ((1),type-modèle,type-modélisation,résultats-modélisation).

.Données originales associées à la surface =

-Si version simple alors

Si type-définition = 'P' alors données = (coordonnées des points) de la surface nom.o.*
appropriée au point de vue de la représentation.

Si type-définition = 'E' alors données = (coefficients des équations) de la surface nom.o

-Si version géographique alors

Si type-définition = 'P' alors données = (coordonnées des points) de la surface nom.o.*
appropriée au point de vue de la représentation.

Si type-définition = 'E' alors données = (coefficients de l'équation) de la surface nom.m.geo.

-Si version chimique alors données = (coordonnées des points) de la surface nom.o.

.N° d'état de la surface = i.

.Liste d'affichage de la surface = (coordonnées-écran du segment graphique de la surface) et (paramètre de visibilité du segment =0 ou 1).

- .Liste ordonnée d'opérateurs de visualisation = (opérateurs de visualisation = (VUCAC) ou (POINT) ou (AXES) ou (ROTAT) ou (SOUS1) ou (INTEN) ou (BLOCD) ou (ISOLI)).
- .Liste ordonnée d'opérateurs de manipulation = (opérateurs de manipulation = (ROT) ou (TRA) ou (FENE) ou (SCA) ou (ACCES) ou (COUPE)).(=LOV).
- .Liste initiale ordonnée d'opérateurs de manipulation = (type-manipulation), de la surface NOM.M.NO.(=LIOM).
- . Pré-conditions : .Si type-définition = 'P' et type-surface = 'SV' alors Nom de surface = (NOM-SURFACE.O.A.NO) ou (NOM-SURFACE.O.M.NO).
- .Type d'opération = (ETUDE 1).
- .Si i = 0 alors liste d'affichage = \emptyset (surface première(non modifiée)et non encore visualisée).
sinon liste d'affichage $\neq \emptyset$ et paramètre de visibilité = 1.
- .Liste ordonnée d'opérateurs de manipulation = \emptyset , si i = 0 et nom de surface \neq NOM.M.NO.
- . " " " " " (LOM) $\neq \emptyset$, si i \neq 0 OU si nom de surface = NOM.M.NO.
- .Liste ordonnée d'opérateurs de visualisation = \emptyset , si i = 0.
- .LOM = Liste initiale d'opérateurs de manipulation(LIOM), si i = 0 et nom de surface = NOM-SURFACE.M.NO.
- . Résultats : .Idem que les arguments.
- .Type d'option d'étude = (MODIFLOOK) ou (MANIP) ou (OUTETUDE).
- .Eventuellement, -surfaces imprimées.
- manipulations mémorisées avec MANIPULATION += (NOM.M.NO;TYPE-MANIPULATION,SURFACE-ORIGINE) (1).
- .Appel au module n° 610, si type d'option d'étude = (MODIFLOOK).
- n° 611, " " " " = (MANIP).
- n° 31, " " " " = (OUTETUDE).
- . Post-conditions : .Liste d'affichage $\neq \emptyset$ et paramètre de visibilité = 1.
- .Si i = 0 alors i = 0
sinon i = i.
- .Si (1) alors nom-surface.m.no = nom de surface,
type-manipulation = LIOM + LOM.
surface-origine = nom de surface.

A III.2 Module n° 615 : Enregistrement renseignements surface.

.Arguments : .Variable de version.

.Type d'opération.

.Pré-conditions : .Variable de version = [0,2].

.Type d'opération = (NEW).

.Résultats : .Si var. de version = 0 alors -Renseignements généraux de la surface SURFACE.O.
-Eventuellement, impression de ces renseignements.
-Si type-définition = 'P' alors appel au module n° 616.
sinon appel au module n° 617.

OU -Appel au module n° 63.

.Si var. de version = 1 alors -Renseignements généraux de la surface SURFACE.O.
-Eventuellement, impression de ces renseignements.
-Le type-définition = 'P' alors appel au module n° 616.

OU -Appel au module n° 66.

.Si var. de version = 2 alors -Renseignements généraux de la surface SURFACE.O.
-Eventuellement, impression de ces renseignements.
-Le type-définition = 'P' alors appel au module n° 616.

OU -Appel au module n° 65.

.Post-conditions : .Renseignements sont validés syntaxiquement et sémantiquement (voir A II.7.2.d à la fonction 1).

.Si var. de version = 1 et si type-centre = 'C' alors calcul de coord-centre par la méthode de Bussière et Stoval (1981, p.103).

.Si var. de version = 1 et si type-centre = 'I' alors introduction de coord-centre par l'utilisateur.

A III.3 Module n° 616 : Enregistrement données surface P.

- ```

. Arguments : .Renseignements généraux d'une surface originale.
 .Var. de version.
 .Fichier de POINTS.

. Pré-conditions : .Type-définition = 'P'.
 .Si var. de version = 0 alors POINT = ((triplet de coordonnées spatiales)*).
 = 1 alors POINT = (si type-données = 'P' alors (triplet de coordonnées spatiales)*,
 = 'D' alors (" " " ")*,
 = 'PD' alors 2(triplet de coordonnées spatiales)*).
 = 2 alors POINT = ((triplet de coordonnées spatiales)* et (triplet de coordon-
 nées spatiales des rayons vecteurs)*).
 .Fichier et renseignements validés et cohérents au point de vue type-surface.
 .Var. de version = [0,2].

. Résultats : .Données originales mémorisées.
 .Renseignements généraux mémorisés.
 .Eventuellement, impression des données originales.
 .Appel au module n° 63 , si var. de version = 0.
 n° 66 , " " " = 1.
 n° 65 , " " " = 2.
OU .Appel au module n° 63 , si var. de version = 0.
 n° 66 , " " " " = 1.
 n° 65 , " " " " = 2.

. Post-conditions : .Données originales validées.

```



A III.4 Module n° 617 : Enregistrement données surface E. (Avec équations cartésiennes du 1er et 2nd degré).

. Arguments : .Var. de version.

.Renseignements généraux d'une surface originale.

.Fichier d'EQUATIONS = ( (équation)\* et type-équation), avec EQUATION =(coefficients)\*,  
avec TYPE-EQUATION =(C1) ou (C2).

. Pré-conditions : .Var. de version =0.

.Type-définition ='E'.

.Fichier et renseignements validés et cohérents au point de vue type-surface.

. Résultats : .Données originales mémorisées.

.Renseignements généraux mémorisés.

.Eventuellement, impression des données originales.

.Appel au module n° 63 ,sivar. de version.=0.

n° 66 , " " " " =1.

n° 65 , " " " " =2.

OU .Appel au module n° 63 ,si var. de version =0.

n° 66 , " " " " =1.

n° 65 , " " " " =2.

. Post-conditions : .Données originales validées.

A III.5 Module n° 611 : Gestion manipulation surface.

- Arguments : .Nom de surface.
  - .Liste d'affichage de la surface.
  - .N° d'état de la surface =  $k' - n'$ .
  - .Type d'option d'étude =(MANIP).
  - .LOM de la surface.
- Pré-conditions :  $.k' - n' \geq 0$ .
- Résultats : .Nom de surface.
  - .Liste d'affichage de la surface .
  - .N° d'état de la surface =  $k'$ .
  - .LOM de la surface.
  - .Si opérateur de manipulation d'état  $x$ :  $k' - n' \leq x \leq k'$  et  $x=(\text{COUPE})$  alors, éventuellement,
    - .coupes imprimées.
    - ET/OU manipulations mémorisées.
- Post-conditions :  $.n' \geq 0$ .
  - .Si 3 manipulation mémorisée alors dernier opérateur de manipulation de type-manipulation =(COUPE).
- Invariant : .Paramètre devisibilité = 1.



A III.6 Module n° 53 : Calcul coordonnées-écran.

• Arguments : .Nom de surface.

.N° d'état de la surface =i.

Si i =0 alors -Renseignements généraux de la surface.

-Données originales de la surface.

-LIOM de la surface.

sinon -Liste d'affichage de la surface.

-Paramètre courant i=( $\emptyset$ ) ou (opérateur de manipulation) ou (opérateur de visualisation).

-Type de MAJ de la surface = (ADD) ou (SUP) ou (REG).(addition ou suppression d'un opérateur régénération de la surface première).

-LOM de la surface.

-LOV de la surface.

• Pré-conditions : .Si n° d'état i =0 et nom de surface = NOM-SURFACE.M.NO alors

Si dernier opérateur de type-manipulation = (COMP-SPAT) alors données originales sont associées aux deux surfaces de surface-origine.

sinon données originales sont associées à la surface de surface-origine.

.Si n° d'état i =0 et nom de surface  $\neq$  NOM-SURFACE.M.NO alors

LIOM  $\neq \emptyset$ .

.Opérateur de manipulation i  $\neq$  opérateur de manipulation i-1.

.Opérateur de visualisation i  $\neq$  opérateur de visualisation i-1.

.Paramètre courant est valide : .par rapport aux renseignements de la surface.

.Si type de MAJ =(REG) alors paramètre courant = $\emptyset$ .

• Résultats : .Nom de surface.

.N° d'état j de la surface.

.Liste d'affichage de la surface.  
 .LOM de la surface.  
 .LOV de la surface.  
 .Variable de calcul "vu et caché" et "comparaison statistique".

. Post-conditions : .Si  $i = 0$  alors -  $j = 1$ .

-LOM = LIOM.

-LOV =  $\emptyset$ .

-Variable de calcul "vu et caché" et "comparaison statistique" = 0.

sinon -si paramètre courant  $\neq$  (VUCAC) ou (COMP-STAT) alors

\*si type de MAJ de la surface =(ADD) alors  $j=i+1$ .

" " " " " " " =(SUP) "  $j=i-1$ .

" " " " " " " =(REG) "  $j=1$ .

\*si paramètre courant = opérateur de manip. alors LOM =LOM + paramètre courant.  
 type de MAJ =(ADD)

\*si paramètre courant = opérateur de visu. alors LOV=LOV + paramètre courant.  
 type de MAJ =(ADD)

\*si paramètre courant = opérateur de manip. alors LOM =LOM - paramètre courant.  
 type de MAJ =(SUP)

\*si paramètre courant = opérateur de visu. alors LOV =LOV - paramètre courant.  
 typ de MAJ =(SUP)

\*si paramètre courant = $\emptyset$  alors LOM = LIOM et LOV = $\emptyset$ .

\*variable de calcul "vu et caché" et "comparaison statistique" = 0.

sinon

\*si paramètre courant =(COMP-STAT) alors  $j=i$  et variable de calcul "vu et caché"=1.

\* " " " =(VUCAC) alors  $j=i$  et variable de calcul "comp. stat."=1.



A III.7 Module n° 41 : Surface originale (MD).

. A . Structure de donnée.

1. Renseignements généraux.

.Si surface originale alors structure = entité SURFACE ORIGINALE associée à EQUATION par DEFINIE-PAR-1.

OU à POINT par DEFINIE-PAR-2.

ET, si type-définition = 'P', SURFACE ORIGINALE APPROXIMÉE associée par ORIGINE-DE-1.

OU SURFACE ORIGI. MATRICIELLE associée par ORIGINE-DE-2.

.Si surface originale géographique alors structure = SURFACE ORIGINALE associée à POINT par DEFINIE-PAR-2.

ET, SURFACE ORIGINALE APPROXIMÉE associée par ORIGINE-DE-1.

OU SURFACE ORIGINALE MATRICIELLE associée par ORIGINE-DE-2.

ET, APPLICATION GEOGRAPHIQUE associée par ASSOCIE- A '-1.

.Si surface originale chimique alors structure = SURFACE ORIGINALE associée à POINT par DEFINIE-PAR-2.

ET, APPLICATION CHIMIQUE associée par ASSOCIE-A-2.

2. Données originales.

.Si surface originale alors structure = entité EQUATION associée à SURFACE ORIGINALE par DEFINIE-PAR-1.

OU entité POINT associée à SURFACE ORIGINALE APPROXIMÉE par DEFINIE-PAR-3.

OU à SURFACE ORIGINALE MATRICIELLE par DEFINIE-PAR-4.

.Si surface originale géographique alors structure = entité POINT associée à SURFACE ORIGINALE APPROX. par DEFINIE-PAR-3.

OU à SURFACE ORIGINALE MATRI. par DEFINIE-PAR-4.

.Si surface originale chimique alors structure = entité POINT associée SURFACE ORIGINALE par DEFINIE-PAR-1.

### B . Opérations sur la structure.

On proposera à l'utilisateur les combinaisons d'opérateurs suivantes:

|           |                |                                |
|-----------|----------------|--------------------------------|
| LIRE      | RENSEIGNEMENTS | SURFACE ORIGINALE              |
| ECRIRE    |                | SURFACE ORIGINALE GEOGRAPHIQUE |
| SUPPRIMER | DONNEES        | SURFACE ORIGINALE CHIMIQUE     |
| TEST D'3  |                |                                |

La spécification de ces opérateurs est simple. Nous donnons simplement quelques exemples:

• Lire données (nom de surface originale).

- Arguments : .Nom de surface.
- Pré-conditions : .Nom de surface.
- Résultats : .Données originales associées à (nom de surface originale)
- Post-conditions : .Données sont valides.

•Ecrire renseignements (nom de surface originale chimique).

- Arguments : •Renseignements généraux associés à (nom de surface originale chimique).
- Pré-conditions : •Renseignements sont valides.  
•Nom de surface originale chimique.
- Résultats : •Mémorisation des renseignements.
- Post-conditions : •Nom de surface originale chimique.

• Test d'existence données (nom de surface originale géographique).

- Arguments : .Nom de surface originale géographique.
- Résultats : .Booléen d'existence.
- Post-conditions : .Si nom existe alors booléen = vrai.  
sinon booléen = faux.



### A III.8 Module n° 44 : Saisie et validation syntaxique.

#### Remarque importante.

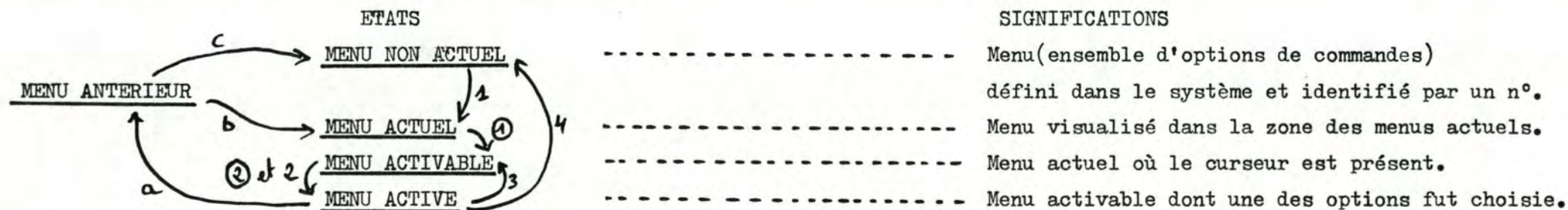
1. Le module Validation Sémantique (N° 45) n'est pas implémenté car <sup>on</sup> supposera une introduction correcte des informations à valider sémantiquement.
2. Les fonctions de saisie au clavier et de validation syntaxique étant fort nombreuses, on spécifiera ici la philosophie d'une saisie et validation. Lors de la mise en oeuvre des modules (algorithmes de conception), on détaillera ces fonctions.

- Arguments : .N° de fonction de saisie et de validation.
- Pré-condition : .N° de fonction valide.
- Résultats : .Données associées au n° de fonction.
- Post-condition : .Données saisies et validées syntaxiquement.

### A III.9 Module n° 410 : Gestion zone menu actuel.

#### Remarques importantes :

1. Afin de mener une gestion intelligente des menus au sein des zones de menu actuel et de menu antérieur, nous avons établi le graphe suivant des états d'un menu:



①, ② : Changements d'état gérés par le module n° 412.

a, b, c : " " " " " " n° 411.

1, 2, 3, 4 : " " " " " " n° 410.



2. Afin de spécifier les modules n° 410 et 412, nous avons dû procéder à une schématisation du déroulement d'une interaction, tout en respectant le graphe des états des menus.

Voici donc les étapes d'une interaction par le biais d'une commande au menu actuel. C'est ce type d'interaction qui nous intéresse pour spécifier les modules n° 410 et 412. La gestion des menus antérieurs est simplement citée (le module n° 411 n'est en effet pas spécifié).

Le module "maître" (qui active les modules n° 410 et 412) sera un module du niveau 6 ou, si l'interaction est courante, le module n° 51.

Décrivons le séquençement des étapes:

1. Activation du module n° 410 en exportant le n° d'interaction à déclencher (C=1).

2. Le module n° 410 affiche le menu associé à l'interaction et exporte au module n° 412 les coordonnées du lieu d'affichage ainsi que le nombre d'options du menu.

3. Activation du module n° 412 pour visualiser le curseur (A=1).

4. Activation du module n° 412 pour activer le curseur (A=2) et déclencher le polling de celui-ci. On exportera le n° d'option vers le module n° 410.

5. Si import d'un n° d'option alors activation du module n° 410 pour allumer l'item correspondant (C=2).

6. Si changement d'interaction alors -Activation du module n° 410 pour supprimer le menu (C=4). et  
" " " n° 412 " " " curseur (A=3).

OU -Activation du module n° 411 pour placer le menu dans la zone des menus ant.  
" " " n° 410 pour supprimer le menu actuel (C=4) et  
" " " n° 412 " " " curseur (A=3) et

Retour en 1.

sinon Attente de la fin d'option et

Activation du module n° 410 pour éteindre l'option allumée (C=3) et

Activation " " n° 412 " supprimer le curseur (A=3) et

Retour en 3.

. Arguments : .N° d'interaction =A.

.N° d'item =B.

.N° de fonction de gestion =C.



- . Pré-conditions : .Si  $C=1$  alors  $B=0$  et  $A \neq 0$  et zone des menus actuel est vide.
  - .Si  $C=2$  alors  $B \neq 0$  et  $A \neq 0$  et menu de A est actuel. et l'item B est éteint.
  - .Si  $C=3$  alors  $B \neq 0$  et l'item B est allumé et  $A \neq 0$  et menu de A est activé.
  - .Si  $C=4$  alors  $B=0$  et  $A \neq 0$  et menu de A est activé..
- . Résultats : .La zone des menus actuels est mise à jour suivant C:
  - Si  $C=1$  alors. le menu de l'interaction A est visualisé dans la zone et devient actuel et,
    - .coordonnées-écran des limites du menu et nombre d'options du menu.
  - Si  $C=2$  alors le menu de l'interaction A comporte un item B allumé.
  - Si  $C=3$  alors " " " " " " " " " éteint.
  - Si  $C=4$  alors " " " " " " " est supprimé et devient non-actuel.

#### A III.10 Module n° 412 : Gestion souris.

- . Arguments : .N° de fonction de gestion =A.
  - .N° d'options activables =B.
  - .Tableau de gestion d'un menu = (4 coordonnées des sommets de la zone de visualisation du menu, nombre d'options).
  - =C.
- . Pré-conditions : .Si  $A=1$  alors le curseur est invisible et le tableau de gestion  $\neq 0$  et  $B=0$ .
  - .Si  $A=2$  alors le curseur est visible et  $B \neq 0$  et  $C=0$ .
  - .Si  $A=3$  alors le curseur est visible.
- . Résultats : .Si  $A=1$  alors le curseur est visualisé dans la zone C.
  - .Si  $A=2$  alors -le curseur est activable dans les options B.(début du polling).
    - Si activation du curseur alors n° d'option choisie
      - le curseur devient inactivable.(fin du polling).
  - .Si  $A=3$  alors le curseur est rendu invisible dans la zone C.

- . Post-conditions : . Si il n'y a pas d'activation du curseur alors le polling continue jusqu'à la fin d'une tranche de temps.

#### A III.11 Module n° 413 : Gestion zone surface.

- . Arguments : . N° de fonction de gestion = A
  - . Liste d'affichage segmentée d'un objet.
  - . Nom du segment à afficher.
- . Pré-conditions : . Si A = 1 alors objet non encore visualisé
  - liste d'affichage  $\neq \emptyset$ , paramètre de visibilité = 0.
  - . Si A = 2 alors objet visualisé
    - liste d'affichage  $\neq \emptyset$ , paramètre de visibilité = 1.
  - . Si A = 3 alors objet visualisé
    - liste d'affichage  $\neq \emptyset$ , paramètre de visibilité = 1.
- . Résultats : . Si A = 1 alors objet visualisé (paramètre de visibilité = 1).
  - . Si A = 2 alors objet rendu invisible ( paramètre de visibilité = 0)
  - . Si A = 3 alors segment de l'objet est supprimé

#### A III. 12 Module n° 414 : Gestion boîte fonctions.

- . Arguments : . N° de fonction de gestion = A.
  - . N° de boutons à gérer = B
  - . N° de potentiomètres à gérer = C
- . Pré-conditions : . Si A = 1 alors C = 0, B  $\neq$  0
  - . Si A = 2 alors C  $\neq$  0 et B = 0.
  - . Si A = 3 alors C  $\neq$  0 et B = 0.



- . Résultats : . Si A = 1 alors - les boutons de n° B sont allumés et activés (le polling est amorcé).
  - si activation, alors renvoi du n° de bouton pressé.
  - désactivation de l'ensemble des boutons.
- . Si A = 2 alors - les potentiomètres de n°C sont activables ( le polling est amorcé).
  - si activation alors renvoi du n° du potentiomètre activé et de la valeur de rotation.
- . Si A = 3 alors les potentiomètres de n°C sont inactivables.

#### A III.13 Module n° 415 : Gestion table de digitalisation.

- . Arguments : . N° de fonction de gestion = A.
  - . Paramètres d'activation de la table : - résolution .
    - écho du curseur.
    - dimension de la table.
- . Pré-conditions : . Si A = 1 alors table inactivable et curseur invisible.
  - . Si A = 2 alors table inactivable et curseur visible (selon une forme n°1).
  - . Si A = 3 alors table inactivable et curseur visible (selon une forme n°1).
- . Résultats : . Si A = 1 alors table inactivable et curseur visible (selon une forme n°1).
  - . Si A = 2 alors-table activable (le polling est amorcé)
    - si activation alors.renvoi de la coordonnée (X,Y) exprimée en termes du système d'axes de la table,
    - .le curseur pendant l'activation se présente sous la forme n°2.
    - désactivation du curseur et retour à la forme n°1 de celui-ci.
  - . Si A = 3 alors la table est inactivable et le curseur est invisible.

#### A III 14 Module n° 416 : Gestion affichage résultats.

Remarque : Ce module pourrait gérer la production de résultats sur différents supports autres que l'écran annexe :  
support papier, support magnétique ...

- . Arguments : . En général : - fichier de données  
- format d'affichage  
- support requis

- . En particulier : -(coordonnées écran) \* sur support papier
  - renseignements généraux sur écran annexe ou sur support papier
  - données originales sur écran annexe ou sur support papier
  - trames d'enregistrement des renseignements généraux et des données sur écran annexe.
  - messages des zones de l'écran annexe
  - ...

- . Pré-conditions : . Validité des données du fichier.  
. Validité du support de production.

- . Résultats : Les données sont produites suivant les spécifications de format et de support.



## ANNEXE IV

Spécification de l'interface.

Remarques: a) Nous allons ici spécifier l'interface des modules n°69,611 du niveau 6.

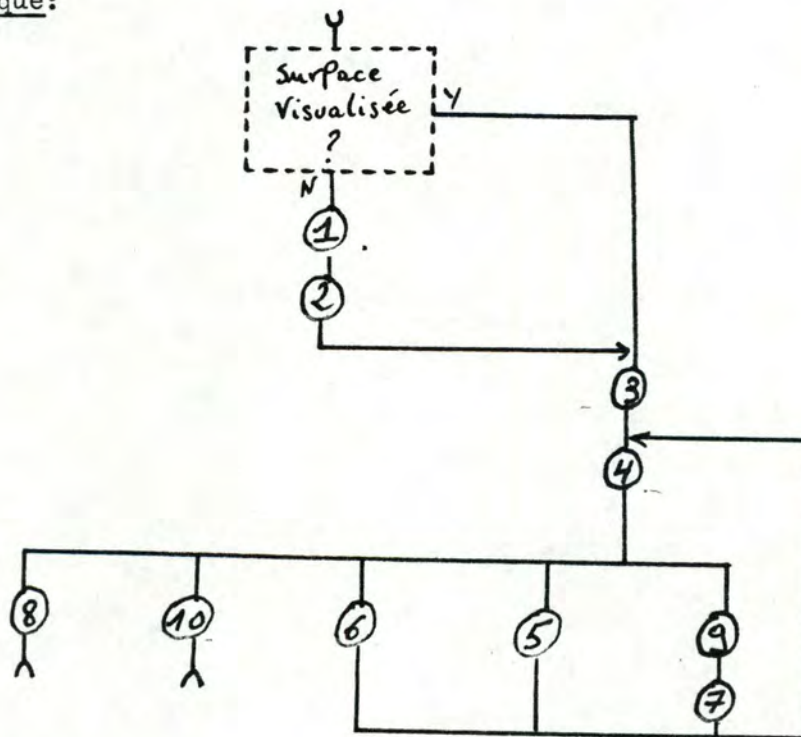
b) Pour la gestion des la zone des Menus Antérieurs,nous avons volontairement omis de la préciser.En effet,celle-ci est dépendante des choix de l'utilisateur propres à sa session de travail.Dans le futur,il serait bon de spécifier un "graphe"reprenant tous les chemins possibles d'utilisation au sein du logiciel.Chaque chemin correspondant alors à une succession de choix,à mémoriser dans la zone des menus antérieurs.

#### IV.1. Module n°69 : Initialisation étude 1 surface.

##### IV.1.1. Task Model

- a. Tâches: 1: Calcul des coordonnées-écran (non-interavtive -NI)  
 2: Affichage de la surface à étudier (NI)  
 3: Affichage du menu (NI)  
 4: Choix d'une option (interactive - I)  
 5: Mémorisation d'une surface (I)  
 6: Impression d'une surface (I)  
 7: Affichage de la surface régéhérée (NI)  
 8: Sortie de la phase d'etude d'une surface (NI)  
 9: Régénaration de la surface(pour obtenir la surface première) (NI)  
 10: Sortie de la phase d'initialisations.(NI).

b. Dynamique:



c. Tâches non-interactives : "Information display" et "Feedback".

- 1: . Zone de titre : "Initialisation de l'étude de la (nom-surface)"  
est affiché.
  - . Zone de feedback : "je calcule la surface à l'étude "est  
affiché.
  - . Zone d'erreur : Si erreur lors du calcul " erreur de calcul  
des coordonnées " est affiché.
- 2: . Zone de titre : Voir 1.
  - . Zoned de surface : affichage de la surface.
- 3: . Zone de titre : Voir 1.
  - . Zone de menu actuel : affichage des options suivantes:  
MANIP.  
LOOK.  
REGEN.  
PRINT  
MEMO.  
OUT.  
HELP.  
IGNORE  
  
Le curseur est présent dans la zone.
    - . Zone de statut : "Voulez-vous choisir une des options sui-  
vantes?" est affiché .
    - . Zone de surface: Surface à l'étude affichée.
- 7: . Zone de titre : Voir 1.
  - . Zone de menu actuel : l'item "REGEN " est allumé  
le curseur est inactivable dans la  
zone, sauf pour IGNORE
    - . Zone de surface: Surface première est affichée.
- 8: . Zone de surface: Surface à l'étude est affichée.
- 9: . Zone de titre : Voir 1.
  - . Zone de menu actuel : l'item "REGEN " est allumé.  
le curseur est inactivable sauf pour  
l'item "IGNORE
- 10: . Zone de titre : Voir 1.
  - . Zone de menu actuel, l'item "MANIP" ou "LOOK" est allumé.
  - . Zone de surface : surface à l'étude affichée.



d. Tâches interactives : Objets manipulables et Actions.

4 : OBJETS intrinsèques : option du menu affiché.

de contrôle : Zone d'item, souris et son curseur.

ACTION : choisir une option du menu.

5 : OBJETS intrinsèques : surface à l'étude.

de contrôle : boutons n°1,2 de la boîte à fonctions.

ACTION : mémoriser ou non la surface.

6 : OBJETS intrinsèques : surface à l'étude.

de contrôle : boutons n° 1,2 de la boîte à fonctions.

ACTION : imprimer ou non sur surface.

IV. 1.2 Control Model (pour les tâches interactives)

a. Commandes:

4 : . On positionne le curseur de la souris dans une zone d'item du menu actuel. Puis, on presse le bouton associé à la souris pour choisir une des options.

. Condition d'erreurs : - Si mémorisation et que la surface visualisée est première ou est déjà mémorisée (1)

- Si IGNORE et qu'une commande n'a pas été exécutée (2)

- Si "REGEN" et que la surface visualisée est première.(3).

5 : . On presse le bouton allumé pour déclencher la mémorisation de la surface ou pour ne pas la réaliser.

6 : . On presse le bouton allumé pour déclencher l'impression de la surface ou pour ne pas la réaliser.

b. Information Display et Feedback.

4 : . Zone de titre : Voir 1.

. Zone de surface : surface à l'étude affichée.

. Zone du menu actuel : . Si le choix d'option<sup>est</sup> valide, l'item correspondant à la position du curseur est allumé.

. Zone d'erreur : - Si(1), alors "surface déjà mémorisée" est affiché.

- Si(2) alors "commande invalide" est affiché.

- Si(3), alors "surface première " est  
affiché.

5 : . Zone de titre : Voir 1.

- . Zone de surface : surface à l'étude affichée.
- . Zone de menu actuel : . item "MEMO" allumé.
  - . curseur inactivable dans la zone sauf pour IGNORE
- . Zone de feedback : Si bouton pressé : "J'enregistre la surface (Nom de surface.M.NO+1)".
- . Zone de statut : . "Pour mémoriser la surface, presser le bouton n°1. Sinon, le bouton n°2 est affiché.
  - . Si un bouton est pressé, alors, les boutons sont éteints et le message de statut disparaît.

6 : . Zone de titre : Voir 1.

- . Zone de surface : surface à l'étude affichée.
- . Zone de menu actuel : . item "PRINT" allumé.
  - . curseur inactivable dans la zone sauf pour IGNORE
- . Zone de feedback : Si bouton pressé, " J'imprime la surface (nom de surface)" est affiché.
- . Zone de statut : . "Pour imprimer la surface, presser le bouton n°1. Sinon le bouton n°2" est affiché.
  - . Si un bouton est pressé alors les boutons sont éteints et le message disparaît.

#### A IV.2. Module n°611 : Gestion Manipulation Surface.

##### A IV.2.1. Task Model.

a. Tâches. 1: Affichage du menu (NI)

2: Restitution de la surface avant la dernière coupe ou fenêtre.(N.I)

3: Sortie de la phase de gestion manipulation (NI)

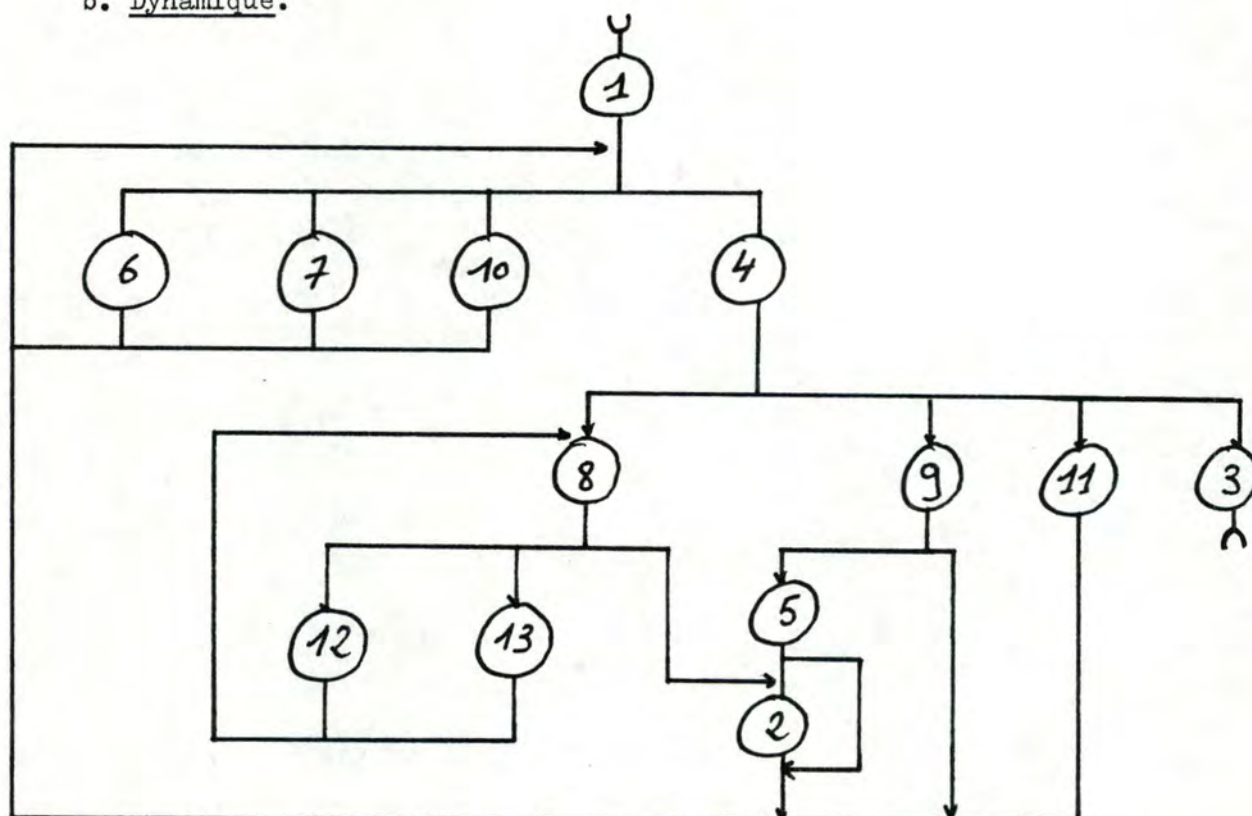
4: Choix de la manipulation non-hardwair à effectuer,(I)

5: Choix du retour à la surface avant le dernier "fenêtrage" (I).



- 6: Effectuer une rotation d'une surface.(I)
- 7: Effectuer une translation d'une surface.(I)
- 8: Effectuer une coupe dans la surface.(I)
- 9: Effectuer un "fenêtrage" d'une surface.(I)
- 10: Effectuer un changement d'échelle d'une surface.(I)
- 11: Accéder à des points d'une surface.(I)
- 12: Mémoriser une coupe.(I)
- 13: Imprimer une coupe.(I)

b. Dynamique.



c. Tâches Non-interactives : "Information Display" et "Feedback".

- 1 : . Zone de titre: "Gestion des manipulations de la (Nom surface)" est affiché.
- . Zone de surface : Surface à l'étude est affichée.
- . Zone de menu actuel : . affichage du menu dont la Zone A est allumée.  
(voir "menu" à la page suivante)
- . le curseur est présent dans la zone B.

|   |        |    |    |   |
|---|--------|----|----|---|
| A | 1      | 2  | 3  | 4 |
|   | RX     | RY | RZ | S |
| B | 5      | 6  | 7  |   |
|   | TX     | TY | TZ |   |
|   | COUPE  |    |    |   |
|   | FENET  |    |    |   |
|   | POINT  |    |    |   |
|   | OUT    |    |    |   |
|   | HELP   |    |    |   |
|   | IGNORE |    |    |   |

- . Zone de statut : "Voulez-vous choisir une des options suivantes, par la boîte fonctions ou le curseur?" est affiché.

2 : . Zone de titre : Voir 1.

- . Zone de surface : surface origine de la coupe ou du "fenêtrage" rétablie.
- . Zone de menu actuel : . zone d'item "coupe" ou "FENET" allumée.
  - . curseur inactivable dans la zone B.
  - . zone A éteinte.
- . Zone de feedback : " je régénère la surface de base" est affiché.

3 : . Zone de surface : surface à l'étude affichée.

d. Tâches interactives : Objets manipulables et Actions.

- 4 : OBJETS intrinsèques : option de manipulation de la surface de contrôle : Zone d'item, souris et son curseur.  
ACTION : choisir une option du menu (zone B).
- 5 : OBJET intrinsèque : surface de base à un fenêtrage.  
de contrôle : bouton à presser n°1 de la boîte fonction.  
ACTIONS : régénérer la surface avant "fenêtrage" ou conserver la fenêtre.



6 : OBJET intrinsèque : surface.

de contrôle : trois potentiomètres (n°1-3) de la boîte fonction.

ACTION : effectuer une rotation d'un angle  $x$  autour d'un axe donné  $(X,Y,Z)$ .

7 : OBJET intrinsèque : surface.

de contrôle : trois potentiomètres (n°5-7) de la boîte fonction.

ACTION : effectuer une translation de la surface le long d'un axe donné  $(X,Y,Z)$

8 : OBJETS intrinsèques : surface, plan de coupe;

de contrôle : clavier, souris et son curseur, zone d'item.

ACTIONS : . définir un plan de coupe,

. réaliser la coupe,

. mémoriser la coupe,

. imprimer la coupe,

. régénérer la surface de base à la coupe.

9 : OBJET intrinsèque : surface.

de contrôle : souris et son curseur, zone d'item.

ACTIONS : . définir une fenêtre.

. réaliser le fenêtrage.

10 : OBJET intrinsèque : surface.

de contrôle : bouton potentiomètre n°4 de la boîte fonction.

ACTION : effectuer un changement d'échelle de la surface.

11 : OBJETS intrinsèques : surface, point d'une surface.

de contrôle : souris et bouton, zone d'item.

ACTION : accéder à la coordonnée d'un point de la surface.

12 : OBJET intrinsèque : coupe de la surface.

de contrôle : bouton n°1 de la boîte.

ACTION : mémoriser une coupe de la surface.

13 : OBJET intrinsèque : coupe de la surface.

de contrôle : bouton n°2 de la boîte.

ACTION : imprimer une coupe de la surface.

#### A IV.2.2. Control Model (pour tâches interactives).

##### a. Commandes :

4 : On positionne le curseur de la souris dans une zone d'item et on presse le bouton associé à cette souris pour choisir une manipulation non-hardware.

5 : On presse le bouton n°1 pour régénérer la surface de base à un fenêtrage.

On presse le bouton n°2 pour conserver la fenêtre.

6 : On tourne le potentiomètre n°1,2 ou 3 pour réaliser respectivement une rotation autour de X,Y ou Z.

7 : On tourne le potentiomètre n°5,6 ou 7 pour réaliser respectivement une translation sur X,Y ou Z.

8 : . On introduit au clavier de l'écran annexe trois points définissant le plan de coupe. On valide cette introduction.

. On introduit "STOP" au clavier pour invalider les trois points et revenir au menu de gestion des manipulations.

. On introduit "OK" au clavier pour valider les trois points et réaliser la coupe. Puis, on positionne le curseur de la souris dans une zone d'item et on presse le bouton associé pour

Mémoriser

Imprimer la coupe de la surface.

Régénérer la surface de base.

9 : On pointe 4 points avec le curseur (validation en poussant le bouton associé). Si un groupe de 4 points est validé on choisit avec la souris une zone d'item pour invalider les 4 points et revenir au menu de gestion des manipulations (STOP).

valider les 4 points et réaliser la fenêtre (OK).

10 : On tourne le potentiomètre n°4 pour réaliser le changement d'échelle.

11 : On positionne la souris sur un point et on presse le bouton associé pour accéder à ce point.

On positionne la souris sur une zone d'item (STOP) pour arrêter la tâche.



12 : Voir tâche n°5 du module n°69 (AIV.1.1.d).

13 : Voir tâche n°6 du module n°69 (AIV.1.1.d).

b. "Information Display" et "Feedback".

4 : . Zone de titre : Voir 1.

. Zone de surface : surface à l'étude affichée.

. Zone de menu actuel : Si choix d'option valide, la zone d'item correspondante à la position du curseur est allumée.

5 : . Zone de titre : Voir 1.

. Zone de surface : surface manipulée affichée.

. Zone de menu actuel : .L'item "FENET" est allumé; la zone A est éteinte.

.Le curseur est inactivable.

. Zone de statut : "Voulez-vous presser le bouton n°1 pour régénérer la surface globale, le bouton n°2 pour conserver la surface "fenêtrée" est affiché.

6 : . Zone de titre : Voir 1.

. Zone de surface : surface en cours de manipulation(rotation) est affichée.

. Zone de menu actuel : . Zone B est éteinte.

. Zone A est allumée.

7 : . Zone de titre : Voir 1.

. Zone de surface : surface en cours de manipulation(translation) est affichée.

. Zone de menu actuel : . Zone B est éteinte.

. Zone A est allumée.

8 : . Zone de titre ; Voir 1.

. Zone de surface : Surface de base est affichée. Puis la coupe est produite à l'écran si la commande "OK" a été introduite.

. La zone A est éteinte

. Zone de menu actuel : . La zone d'item "COUPE" est allumée.

. Le curseur est inactivable sauf dans la zone IGNORE. Jbr 2/85

Quand le plan de coupe est défini, validé  
(commande "OK" au clavier) et affiché. On affiche  
les trois options suivantes : MEMO

PRINT

REGEN

- . On active le curseur pour ces trois options.
- . Si un choix d'option valide est effectué, la zone d'item correspondante à la position du curseur est allumée.
- . Zone de statut : Au départ, on affiche "Voulez-vous travailler à l'écran annexe!"  
Quand la coupe est produite, le message est remplacé par "Voulez-vous choisir une option de gestion de la coupe?"
- . Zone de Feedback : Quand la coupe est calculée, on affiche  
"Je calcule la coupe de la surface (nom surface)  
suivant le plan défini par (coordonnées des 3 points).

Pour la définition d'un plan de coupe :

- . Zone de travail (écran annexe) : Un écran préformaté pour introduire les trois points est affiché.
- . Zone de statut (écran annexe) : "Voulez-vous introduire les trois points du plan de coupe? Taper STOP pour invalider votre introduction. Taper OK pour la valider" est affiché.
- . Zone d'erreur (écran annexe) : Si les trois points ne sont pas validés, on affiche "Plan de coupe invalidé"  
On réitère l'introduction.
- . Zone de feedback : Si les 3 points sont valides, on affiche  
" Je calcule la coupe et je l'affiche à l'écran graphique".

9 : . Zone de titre: Voir 1.

- . Zone de surface : Surface manipulée (définition d'une fenêtre) est affichée. Le curseur y est activable, avant une commande "STOP" ou "OK"



- . Zone de menu actuel : . Zone A, éteinte.
  - . Zone d'item "FENET" est allumée.
  - . Si quatre points sont introduits et validés, on affiche STOP
- OK.
- . Zone de statut : . Tout d'abord, on affiche "Voulez-vous introduire par la souris quatre points de définition de la fenêtre? Choisissez "STOP" pour invalider, "OK" pour valider l'introduction."
- . Zone de feedback : . Après la commande "OK", si quatre points valides sont introduits : "introduction valide" est affiché.
  - . Après avoir choisi un point, une croix est placée au lieu de celle-ci et la coordonnée-écran du plan y est mentionnée (Zone de surface).
- . Zone d'erreur : . Après la commande "OK", si quatre points invalides sont introduits : "introduction invalide" est affiché.

10 : . Zone de titre : Voir 1.

- . Zone de surface : surface en cours de manipulation.
- . Zone de menu actuel : . La zone B est éteinte.
  - . La Zone A est allumée.

11 : . Zone de titre : Voir 1.

- . Zone de surface : surface à l'étude affichée. Le curseur y est activable avant une commande "STOP".
- . Zone de menu actuel : . La zone A est éteinte.
  - . La zone de l'item "POINT" est allumée.
  - . Le curseur est inactivable sauf dans la zone IGNORE. Quand un accès à un point est réalisé, on affiche l'item STOP pour arrêter les accès de points.
  - . on active le curseur dans cette zone et on le désactive de la zone IGNORE.





## ANNEXE V : Conception algorithmique

Remarques : Selon la troisième restriction (voir §VI.1), nous préciserons les algorithmes des modules n°69,53, 410,412 et 413.

Lorsqu'un module A utilise un module B, A exportera vers B les arguments de B et importera éventuellement les résultats de B s'il en a besoin.

A V.1 Module n°69 :Initialisation de l'étude d'une surface.

- Si n°d'état surface = 0, alors

\* Si nom surface = (NOM.o.\*) ou (NOM.M.GEO) alors

. export au module n°53 de : - n° d'état

- nom surface

- liste initiale ordonnée d'opérateurs de manipulation (=  $\phi$ )

- données originales associées.

. import des résultats du module n°53.

sinon

. export au module n°53 de : - n° d'état

- nom surface ( = NOM.M.NO)

- liste initiale ordonnée d'opérateurs de manipulation =  
élément TYPE-MANIPULATION de l'entité NOM.M.NO.

- renseignements généraux associés au nom surface.

- données originales associées au nom surface.

. import des résultats du module n°53.

- \* Export au module n°413 de :
  - nom surface
  - liste d'affichage du segment représentant la surface avec le paramètre de visualisation = 0.
  - n° de fonction de gestion (A = 1)
- Export au module n°410 de :
  - n° d'interaction (A = a)
  - n° d'item (B = o)
  - n° fonction de gestion (C = 1)
- Import des résultats du module n°410.
- Export au module n°412 de :
  - n° de fonction de gestion (A = 1)
  - lieu d'affichage du menu (C)
  - n° d'options activables (B = 0)
- ① - Export au module n°412 de :
  - n° de fonction de gestion (A = 2)
  - lieu d'affichage du menu (C = 0)
  - n° d'options activables (B) : Si n° d'état = 1, alors B = (2,3,4,6)  
sinon B = (1,2,3,4,5,6)
- Import du n° d'option choisie (X = x) venant du module n°412.
- Export au module n°410 de :
  - n° d'interaction (A = a)
  - n° d'item (B = x)
  - n° de fonction de gestion (C = 2)
- ( Remarque : l'affichage du menu et la gestion du curseur pourraient faire partie d'une séquence d'interface commune, réalisable au sein du module n°51.)



- CASE x of

- 1 : (mémorisation de la surface) : 

|                                                    |
|----------------------------------------------------|
| WRITE (renseignements généraux de la manipulation) |
| Export au module n°57 de ses arguments             |

  - . Import des résultats du module n°57 (éventuel message d'erreur).
  - . Aller en ②.

2 : (impression de la surface) : . Export au module n°416 de : - liste d'affichage du segment de la surface.  
- surface.  
  - . Import des résultats du module n°416 (éventuel message d'erreur).
  - . Aller en ②.

3 : (modification des paramètres de visualisation) : . Export au module n°610 de ses arguments  
- . Aller en 3.

4 : (manipulation de la surface) : . Export au module n°611 de : - nom surface  
- liste d'affichage du segment de la surface.  
- n° d'état actuel.  
- type d'option d'étude ( = MANIP ).  
- liste ordonnée d'opérateurs de manipulation.  
  
- . Aller en ③.

5 : . Export au module n°53 de : - liste d'affichage du segment de la surface manipulée.  
- nom surface  
- n° d'état surface  
- type de mise à jour surface ( = REG )  
- paramètre courant ( = 0 )

- . Import des résultats du module n°53.
- . Export au module n°413 de : - nom surface
  - n° fonction de gestion (A = 3)
- . Export au module n°413 de : - nom surface
  - liste d'affichage du segment de la surface première avec le paramètre de visualisation = 0
  - n° fonction de gestion (A = 1)
- . Aller en ②;
- 6 : . CASE variable de version OF
  - ∅ : Export au module n°63 de ses arguments
  - 1 : Export au module n°66 de ses arguments
  - 2 : Export au module n°65 de ses arguments
- . Export au module n°413 de : - nom surface
  - n° fonction de gestion (A = 3)
- . Aller en 3;
- STOP;
- ② - Export au module n°410 de : - n° de fonction gestion (C = 3)
  - n° d'item (B = x)
  - n° d'interaction (A = a)
- Aller en ①.
- STOP ;
- ③ - Export au module n°410 de : - n° de fonction gestion (C = 4)
  - n° d'interaction (A = a)
  - n° d'item (B = 0)
- STOP;
- END.



A V . 2 Module n°53 : Calcul Coordonnées-écran.

Remarque : La conception de cet algorithme est cruciale. Nous proposons donc ici une première approche qui demandera affinements et corrections.

- Si n° d'état surface = 0 alors

\* Si nom surface = (NOM.O.\*) ou (NOM.M.GEO.NO) alors

. Si type de définition de nom surface = 'E' alors

calcul coordonnées réelles = Matrice Mw = [(Xw,Yw,Zw)\*]

Si type d'équation de l'entité EQUATION = C1 alors

calculer  $Ax + By + Cz + D = 0$  pour

X : x min  $\rightarrow$  x max avec intervalle de Xint

calculer  $Ax + By + Cz + D = 0$  pour

Y : y min  $\rightarrow$  y max avec intervalle de Yint

(A) ( Remarque : on calcule les coordonnées réelles de la surface en prenant le système d'axes de référence direct .

On peut introduire les valeurs de x min, x max, Xint et y min, y max, Yint).

Si type d'équation de l'entité EQUATION = C2 alors

calculer  $AX^2 + A'Y^2 + A''Z^2 + 2BXY + 2B'XZ + 2B''YZ + 2CX + 2C'Y + 2C''Z + D = 0$

pour X : x min  $\rightarrow$  x max avec intervalle de Xint

calculer  $AX^2 + A'Y^2 + A''Z^2 + 2BXY + 2B'XZ + 2B''YZ + 2CX + 2C'Y + 2C''Z + D = 0$

pour Y : y min  $\rightarrow$  y max avec intervalle de Yint.

sinon

Si variable de version = ( $\phi$ ) ou (1) alors

calcul coordonnées réelles = Matrice  $M_w = [(X_w, Y_w, Z_w)^*]$

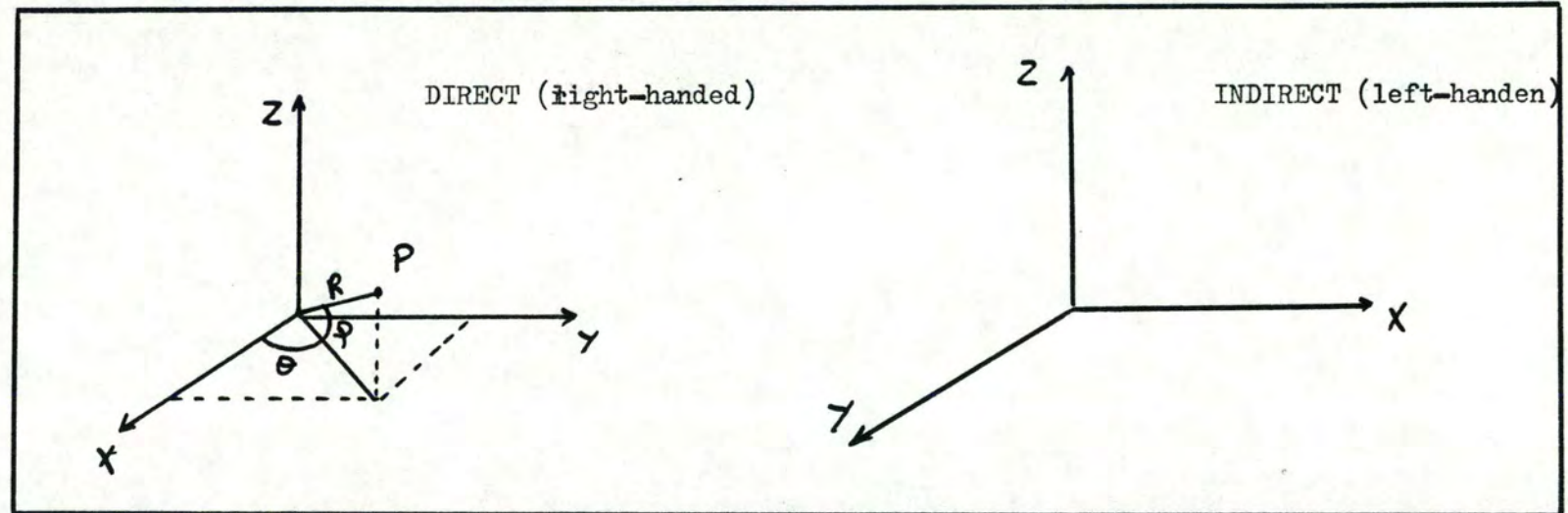
Si nom surface = NOM.D.A alors

calcul de la matrice  $M_w$  à partir des matrices  $P_{ij} F_i(u)$  et  $F_j(v)$  provenant des occurrences de l'entité POINT.

(Remarque : surface =  $\sum_i \sum_j P_{ij} F_i(u) F_j(v)$ )

$M_w = (\text{occurrence de l'entité POINT})$

ⓑ



Réaliser la projection perspective de l'objet ( $M_w$ ) sur l'écran pour obtenir  $M_s$  :

- Calcul de  $M_e = ((X_e, Y_e, Z_e)^*)$  pour appliquer le système d'axes  $(X_o, Y_o, Z_o)$  en  $(X, Y, Z)$ .

ⓒ



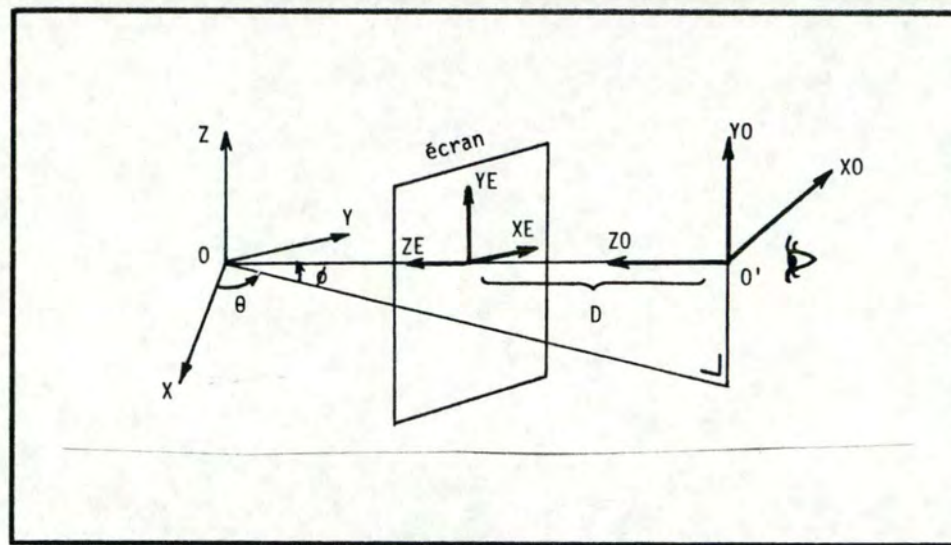


FIGURE 32. Projection perspective

- on peut introduire ici  $\theta$  et  $\varphi$  qui définissent la façon dont est vu l'objet.

$$- \forall (X_w, Y_w, Z_w), X_e = -X_w \sin \theta + Y_w \cos \theta$$

$$Y_e = -X_w \cos \theta \cdot \sin \varphi - Y_w \sin \theta \sin \varphi + Z_w \cos \varphi$$

$$Z_e = -X_w \cos \theta \cdot \cos \varphi - Y_w \sin \theta \cos \varphi - Z_w \sin \varphi + R.$$

○ Eliminer les éléments de l'objet en dehors de la pyramide de visualisation à partir de  $M_e' = ((X_e, Y_e, Z_e, 1)*)$

Cette opération au clipping de l'objet produira  $M_c$ .

$$M_c = M_e \cdot P \text{ avec } P = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & \frac{s}{D(1-\frac{D}{F})} & 0 & 0 \\ 0 & 0 & \frac{-s}{(1-\frac{D}{F})} & 0 & 0 \end{pmatrix}$$

$$\text{avec } M_c = ((X_c, Y_c, Z_c, W)*)$$

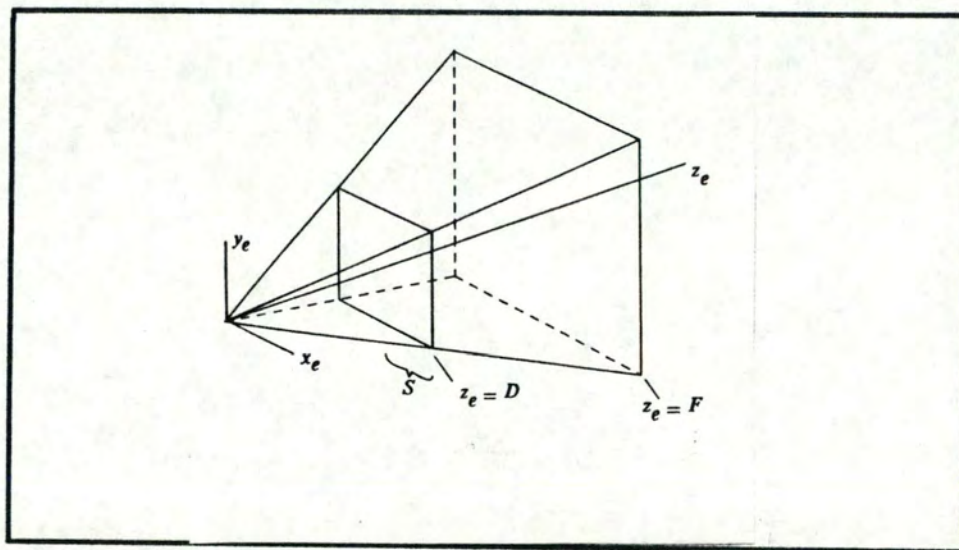


FIGURE 33 : Pyramide de visualisation du clipping.

○ Calculer la matrice des coordonnées écran ( $M_s = [(X_s, Y_s, Z_s)^*]$  )

$$V(X_c, Y_c, Z_c, W), \begin{cases} X_s = \frac{X_c}{W} \\ Y_s = \frac{Y_c}{W} \\ Z_s = \frac{Z_c}{W} \end{cases}$$

©



③

- Créer un segment Move-Draw avec les lignes et colonnes de la matrice  $M_s$  = liste d'affichage.  
(une ligne de la matrice  $M_s$  correspond à une ligne du Wire-frame.  
une colonne de la matrice  $M_s$  correspond à une colonne du Wire-frame).

sinon

- . Pour chaque surface origine de la manipulation (élément SURFACE-ORIGINE de l'entité MANIPULATION identifiée par NOM.M.NO. on effectue les algorithmes
  - Ⓐ si type de définition nom surface origine = 'E'
  - Ⓑ si type de définition nom surface origine = 'P'et on obtient alors 1 ou 2 segments de la ou les surfaces origines de la manipulation.
- . Sur chaque segment, on applique les opérateurs de manipulation de la liste initiale ordonnée d'opérateurs de manipulation.

\* n° d'état de la surface = 1

sinon

\* Modifier le segment d'affichage de la surface en fonction du paramètre courant.

.\_. Si paramètre courant = opérateur de manipulation alors

- CASE paramètre de manipulation OF

\*\* ROT ( $i, \alpha$ ): Si type mise à jour = ADD alors

CASE i OF

Z : réaliser la transformation matricielle de  $M_s'((X_s, Y_s, Z_s, 1)^*)$  par

$$\begin{pmatrix} \cos \alpha & -\sin \alpha & 0 & 0 \\ \sin \alpha & \cos \alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$Y : \text{idem mais par } \begin{pmatrix} \cos \alpha & 0 & \sin \alpha & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \alpha & 0 & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$X : \text{idem mais par } \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha & 0 \\ 0 & \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

(Remarque :  $\alpha$  est mesuré dans le sens des aiguilles d'une montre en regardant vers le centre du système d'axe).

: Si type mise à jour = SUPP alors

CASE i OF

Z  
Y  
X

on utilisera les matrices inverses où  $(\alpha)$  devient  $(-\alpha)$ .

\*\* TRAN ( $T_x, T_y, T_z$ ): Si type mise à jour = ADD alors

réaliser la transformation matricielle de  $M_s'$  par



$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ T_x & T_y & T_z & 1 \end{pmatrix}$$

où  $T_x$ ,  $T_y$ ,  $T_z$  sont les composants de la translation en (X, Y, Z) suivant les trois directives X, Y, Z.

: Si type mise à jour = SUPP alors  
on appliquera la matrice inverse  $(-T_x, -T_y, -T_z)$ .

\*\* SCA ( $S_x, S_y, S_z$ ) : Si type mise à jour = ADD alors  
réaliser la transformation de  $M_s'$  par

$$\begin{pmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

où  $S_x$ ,  $S_y$ ,  $S_z$  sont les facteurs d'échelle dans les trois directions et à partir du centre de l'écran.

: Si type mise à jour = SUPP alors  
on appliquera la matrice inverse  $(-S_x, -S_y, -S_z)$ .

\*\* ACCES( $X_s, Y_s$ ): Si type de mise à jour = ADD alors  
afficher les coordonnées réelles du point (calcul inverse à ③)

: Si type de mise à jour = SUPP alors  
supprimer l'affichage des coordonnées du point  $(X_s, Y_s)$ .

\*\* COUPE  $(X,Y)$  ou  $(X,Z)$  ou  $(Y,Z)$  : Si type de mise à jour = ADD alors  
calculer la coupe à partir de  $M_w$  une nouvelle matrice  $M_{1w}$   
appliquer  $\odot$  à  $M_{1w}$

: Si type de mise à jour = SUPP alors  
supprimer la coupe en exportant au module n°413: nom surface . C  
variable de gestion(A=3)

\*\* FENE  $((X_1,Y_1),(X_2,Y_2),(X_3,Y_3),(X_4,Y_4))$ : Si type de mise à jour = ADD alors  
définir pour le segment de la surface nom surface, une fenêtre.  
: Si type de mise à jour = SUPP alors  
définir pour le segment de la surface nom surface, la fenêtre  
maximale.

. Si paramètre courant = opérateur de visualisation alors

CASE paramètre de visualisation OF

\*\* VUCAC (1): Export au module n°52 de ses arguments; variable de calcul VUCAC=1

\*\* POINT  $((X_w,Y_w,Z_w)*)$ : Si type de mise à jour = ADD alors  
calcul des  $(X_s,Y_s,Z_s)*$  par C  
: Si type de mise à jour = SUPP alors  
supprimer l'affichage des points.



```

** AXES (1) : Si type de mise à jour = ADD alors
 calcul des coordonnées-écran des axes .

** BLOC-D (1) : Si type de mise à jour = ADD alors
** SOUS 1 (1) : calcul des coordonnées-écran de la nouvelle structure
** ISOLI (1) : définir les segments correspondants

 : Si type de mise à jour = SUPP alors
 supprimer les segments créés en exportant au module 413 : - nom du segment
 - variable de gestion (A=3)

** ROT (1) : Si type de mise à jour = ADD alors
** INTE (x) : affecter au segment un attribut de rotation continue ou d'intensité x
 : Si type de mise à jour = SUPP alors
 supprimer cet attribut

 . Si paramètre courant = 0 alors
 réaliser les opérateurs (avec type de mise à jour = SUPP) des listes ordonnées de
 manipulation et de visualisation dans leur ordre inverse.

* Mettre à jour le n° d'état surface
 . CASE type de mise à jour OF
 ADD : n° d'état surface = n° d'état surface +1
 SUPP : n° d'état surface = n° d'état surface -1
 REG : n° d'état surface = 0

```

\* Mettre à jour la liste ordonnée d'opérateurs de manipulation (LOM)

. CASE type de mise à jour OF

ADD : LOM = LOM + paramètre courant.

SUPP : LOM = LOM - paramètre courant

REG : LOM = liste initiale ordonnée d'opérateurs de manipulation (LIOM).

\* Mettre à jour la liste ordonnée d'opérateurs de visualisation (LOV)

. CASE type de mise à jour OF

ADD : LOV = LOV + paramètre courant

SUPP : LOV = LOV - paramètre courant

REG : LOV =  $\emptyset$

#### A V. 3 Module n°410 : gestion Zone Menu actuel.

CASE C OF

1 : . READ (texte-menu) de l'entité INTERACTION identifiée par le n° interaction = A

. Export des arguments au module n° 417.

. Import des résultats du module n° 417.

. Créer un segment du menu actuel avec le paramètre de visibilité = 1 et créer X sous-segments pour chaque option dont le n° : 1  $\rightarrow$  X

. Export au module appelant de : - Coordonnées-écran de la zone d'affichage du menu =  
((X<sub>1</sub>,Y<sub>1</sub>), (X<sub>2</sub>,Y<sub>2</sub>), (X<sub>3</sub>,Y<sub>3</sub>), (X<sub>4</sub>,Y<sub>4</sub>))  
- Nombre d'options de texte-menu (=X)

2 : . Affecter au sous-segment de n° = B, un attribut d'intensité d'affichage élevée.

3 : . Affecter au sous-segment de n° = B, un attribut d'intensité d'affichage normale.

4 : . Supprimer le segment du menu actuel.



A V.4 Module n° 412 : Gestion Souris

CASE A OF

1 : . "enable" du curseur avec écho = 1 dans la zone C

2 : . Tant que pas activation de la souris,

    "sample" du curseur en lisant (x,y); si (x,y) ≠ b alors activation

    . Calcul du n° d'option choisie (X) :  $D = \frac{\text{Surface affichage menu (C)}}{\text{Surface affichage item (E)}}$

    .  $\forall i : 1 \rightarrow D$ , Test si (x,y) ∈ E(i); si oui, X = i.

    . Export au module appelant de X.

3 : "disable" du curseur.

A V 5 Module n° 413 : Gestion Zone Surface.

CASE A OF

1 : Affecter le paramètre de visibilité à 1 du segment de la surface de nom-surface

2 : Affecter le paramètre de visibilité à 0 du segment de la surface de nom-surface

3 : supprimer le segment de la surface de nom-surface.

---